A REPORT
ON

CUSTOMER RELATIONSHIP MANAGEMENT IN SALESFORCE

Submitted by,

Neha H D - 20211CSE0519

Under the guidance of,

Mr. Asad Mohammed Khan

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship report **"Customer Relationship Management in Salesforce"** being submitted by "Neha HD" bearing roll number "20211CSE0519" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Mr. Asad Mohammed Khan	**Dr. Asif Mohamed H B**
Assistant Professor	Assistant Professor & HoD
School of CSE & ISE	School of CSE&IS
Presidency University	Presidency University

Dr. MYDHILI NAIR	**Dr. SAMEERUDDIN KHAN**
Associate Dean	Pro-Vice Chancellor - Engineering
PSCS	Dean –PSCS / PSIS
Presidency University	Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

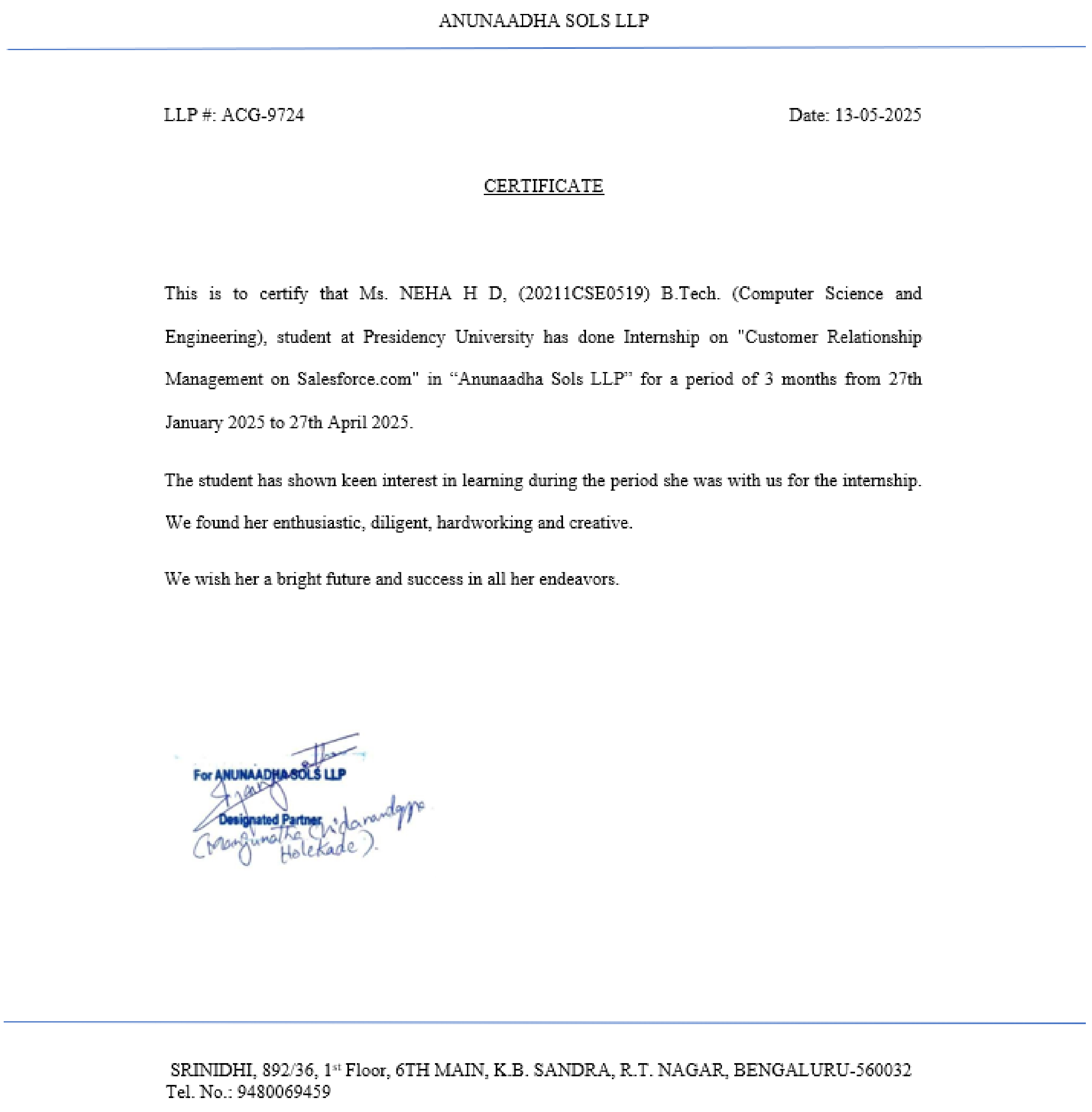
I hereby declare that the work, which is being presented in the report entitled "**Customer Relationship Management in Salesforce"** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of my own investigations carried under the guidance of **Mr. Asad Mohammed Khan, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

Neha H D

20211CSE0519

INTERNSHIP COMPLETION CERTIFICATE



ANUNAADHA SOLS LLP

LLP #: ACG-9724

Date: 13-05-2025

CERTIFICATE

This is to certify that Ms. NEHA H D, (20211CSE0519) B.Tech. (Computer Science and Engineering), student at Presidency University has done Internship on "Customer Relationship Management on Salesforce.com" in "Anunaadha Sols LLP" for a period of 3 months from 27th January 2025 to 27th April 2025.

The student has shown keen interest in learning during the period she was with us for the internship. We found her enthusiastic, diligent, hardworking and creative.

We wish her a bright future and success in all her endeavors.

For ANUNAADHA SOLS LLP

[illegible]

Designated Partner

(Manjunatha Chidanandappa Holekade).

SRINIDHI, 892/36, 1st Floor, 6TH MAIN, K.B. SANDRA, R.T. NAGAR, BENGALURU-560032
Tel. No.: 9480069459

ABSTRACT

CRM is short for Customer Relationship Management. This technology enables you to manage your relationships with your customers and prospects and monitor data on all of your interactions. It also enables teams to work together, both within and outside your organization, listen to social media, monitor key metrics, and communicate through email, phone, social, and other media. By concentrating on the convergence of CRM with sales and marketing strategies, this study illustrates how Salesforce improves productivity, customer retention, and business performance of organizations.

Salesforce gives you everything you need to manage your business from anywhere. With standard products and features, you can run relationships with prospects and customers, work and interact with employees and partners,
and safely store your information in the cloud. But standard products and features are just the starting point.

Salesforce CRM enables companies to develop improved relations, close quicker deals, and deliver outstanding customer experiences. It applies automation and AI features for enhancing customer relations and business performances. The abstract further states that Salesforce is comprehensively applied within various industries thanks to its nature of being customizable and scalable based on the needs of the expanding business.CRM offered in a cloud solution format that automates and simplifies sales, marketing, customer service, and support operations in a company.

ACKNOWLEDGEMENTS

First of all, I indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

I express my sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

I express my heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and "**Dr. Asif Mohamed H B"**, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

I am greatly indebted to our guide **Mr. Asad Mohammed Khan, Assistant Professor** and Reviewer **Dr. Asif Mohamed H B, Head of the Department**, Presidency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

I would like to convey our gratitude and heartfelt thanks to the PIP4004 Internship Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Project Coordinators **Mr. Jerrin Joe Francis** and Git hub coordinator **Mr. Muthuraj.** I thank my family and friends for the strong support and inspiration they have provide us in bringing out this project.

Neha H D

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

1.1 Background and Motivation

Salesforce is your customer success platform, built to help you sell, service, market, analyse, and connect with your customers. Salesforce has all the tools you need to run your business from anywhere. With standard products and capabilities, you can handle relationships with prospects and customers, collaborate and interact with employees and partners, and keep your data safe in the cloud If your data resides in spreadsheets, buried in emails or text messages, or stuck to your bulletin board, it's difficult to get the complete picture of your prospective customer. And you definitely can't view the data from anywhere, anytime, nor view your progress on deals in process, at least not without having to call you for a status check.

When it's all in Salesforce, you don't have to think about the note on your desk, or a file you've got on your hard drive. You can get anything you need by simply logging in. Moreover, you no longer have to worry about keeping your manager in the loop about how things are progressing with your top deals in flight. Instead, your manager can just log into Salesforce and see the latest data in real time. This visibility is one of the key benefits of Salesforce. But rest assured, there are powerful security and sharing features that protect sensitive data and ensure the right folks see the right data. Access determines your ability to open and interact with data stored in Salesforce. The information you can create, read, update, and delete is defined by settings your admin has (more on your admin in the next unit). Access may be single- or multi-layered, depending on how complicated your company's needs are.

The key point you should realize is that Salesforce has controls around who can see and edit data, and your admin assists with setting those up and keeping them running. As long as your security is correctly enabled, your entire company can be on Salesforce, and one of the most compelling reasons to do so is to tap the power of collaboration. Instead of key insights and answers to critical questions residing in individual email inboxes or in the hallway, collaboration in Salesforce is searchable and accessible. This way, your collaboration in Salesforce can be your corporate memory, enabling you to capture and share useful content that accumulates over time, growing in value the more your entire company participates. Adding the Google Maps API enables capabilities like geolocation-based filtering, where users can find appropriate places in their local area or a defined region, providing a vital layer of useful functionality.

Finally, this research aims to overcome the shortcomings of traditional recommendation methods by developing an interactive, smart, and user-friendly web program. The goal is to design a system that, apart from making it easy to identify new destinations, greatly improves the choice-making process for a wide range of users, such as tourists, commercial travelers, and local populations, by offering informative, personalized, and readily accessible suggestions.

1.2 Problem Statement

A product-based organization wants to have a Lead and Opportunity management CRM. They have approximately 100 customers now and would be adding more as they expand. They would need to have a CRM that would enable them to generate Leads (with minimal information captured) and when the customer is ready to proceed, Opportunities can be generated. Company will have a Sales group of 5 + VP + President. Additionally, a System Administrator will be there. All Accounts need to be ONLY visible to everyone but Opportunities and Leads need to be viewable ONLY to the owners. VP and President would have extra permissions on all data.

1.3 Scope of the Project

CRM means Customer Relationship Management. With this technology, you can keep track of your customer and prospect relationships and follow information about all your interactions. It also facilitates collaboration between teams internally and externally, listening to social media insights, monitoring key metrics, and messaging through email, phone, social, and more. Objects in Salesforce are accessed through the navigation menu. Choose any record to drill down to a particular account, contact, opportunity, or any other record in Salesforce. you can both access from your desktop and your mobile phone. So, it's like a spreadsheet, but in Salesforce, your data are all tracked, shared, and have apps associated with them. Salesforce has some standard objects pre-configured and ready to go places, creating personalized opportunities are qualified leads to purchase. No matter how you get there — from Home, the navigation menu, Search, or a related record in Salesforce — when you open an opportunity, you need a robust workspace where you can get things done fast and direct your energy at selling. when a lead is converted, a contact and account are also created in Salesforce. An account is a company you're doing business with, and a contact is someone who works at that account. Like with opportunities, whenever you drill down into an account or contact, you have to find what you need in a hurry.

1.3 Research Significance

The relevance of this research comes from its contribution in the improvement of the field of place recommendation systems by the application and incorporation of machine learning techniques into a contemporary web development environment using Firebase for scalable and adaptable data handling. Admins do actually administer Salesforce using a unique set of permissions that exist only for system administrators. They can configure and tailor your org according to your company's requirements. Admin mastermind Individual creating awesome things with Salesforce. The job title is Salesforce Admin, but for so many admins within our community, they're doing way more than merely administering Salesforce. They're becoming your company's go-to Salesforce advisor. Admins also do administer Salesforce, working with a special permission set reserved for system administrators. They can configure and tailor your org to fit your company's requirements. Depending on the size of your company and the level of complexity in your Salesforce implementation, how your company is handling your Salesforce org may be different. You may have multiple individuals supporting Salesforce, or you may have a single admin, and that individual may even be doing more than administering Salesforce.

Chapter 2
LITERATURE SURVEY

2.1 Introduction to Recommendation Systems

Recommendation systems are now a standard feature of contemporary digital platforms, acting as advanced information filtering mechanisms that aim to forecast user interests and recommend items likely to be of interest. The main goal of these systems is to mitigate the issue of information overload, allowing users to effectively find relevant content, products, services, or, in this case, locations, from a huge set of possibilities [1]. Their widespread adoption across e-commerce, media streaming, social media, and location-based services reflects their worth in enriching user experience, promoting engagement, and enhancing platform utility.

Generally, recommendation systems can be classified into various types depending on the data and methods they use. Collaborative Filtering (CF) systems provide recommendations by exploiting the preferences and actions of a group of users, with the belief that users who concurred in the past will concurs in the future [2]. Content-Based (CB) systems, however, suggest items that are comparable to previously liked items by a user, depending on attributes or properties of the items [2]. Hybrid systems employ multiple techniques for making recommendations to overcome the shortfall of using single methods, providing often more consistent and accurate recommendations [2]. There are other categories such as knowledge-based systems, demographic-based systems, and context-aware systems, which involve contextual data such as time, place, and mood.

Although they have achieved widespread success, recommendation systems have a number of well-documented issues. The "cold start" problem arises when there are new users or new items with little interaction data, and hence it is hard to provide good recommendations [3]. Data sparsity, when the user-item interaction matrix is extremely sparse, can negatively affect the performance of collaborative filtering. Scalability is also a critical concern, as the computational power required to process large data and generate recommendations can grow exponentially with the item base and user base [3]. Moreover, ensuring serendipity (finding novel but pertinent items), ensuring fairness and transparency, and making explainable recommendations are ongoing research and development topics [4].

2.2 Place Recommendation Systems: Specific Challenges and Approaches

Point-of-interest recommendation systems, a narrow specialization of recommendation systems, aim to suggest points of interest (POIs) like restaurants, cafes, parks, tourist spots, and businesses to users. Though they employ basic concepts similar to overall recommendation systems, suggesting places brings specific challenges mainly because of the spatial and temporal nature of location data along with mobility patterns of users.

One of the main challenges is integrating the geographical context. User preferences for locations are strongly determined by their location, distance, and spatial relationships between locations [4]. Recommendations need to be geographically contextual, often favoring nearby or easily reachable locations. Standard recommendation models may fail to consider this important spatial aspect. The temporal context is also crucial; the appropriateness of a place suggestion can be based on the time of day (e.g., suggesting a breakfast place in the morning), day of the week, or season. User behavior for place visitation also has strong temporal patterns.

In addition, user mobility and the sequence of visited locations offer rich, but complex, data. Analyzing user trajectories and sequential patterns can provide insights into common travel behaviors and likely next destinations, but need sophisticated modeling methods [5]. The heterogeneity of places (from utilitarian services to entertainment sites) and different levels of user involvement (a brief visit vs. an extended stay) also introduce data heterogeneity issues [6].

2.3 Application of Platform as a Service (PAAS) in Recommendation Systems

PaaS (Platform as a Service) is a cloud model that offers an existing platform for developers to develop, test, and deploy applications—without worrying about the underlying infrastructure or hardware. PaaS offers an integrated development environment in the cloud. It enables developers to concentrate on coding and not on infrastructure. Users have access to the platform via a web browser, from anywhere. PaaS supports a variety of programming languages and development tools. It facilitates quicker time-to-market through easier application deployment.

- In the context of PAAS Here's the step-by-step explanation of how it works:

  Cloud Provider Hosts the Platform: In PaaS (Platform as a Service), the cloud provider is the service provider or company (such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, or Salesforce) that owns and operates the whole computing infrastructure needed to execute applications. The cloud provider manages all physical hardware—servers, storage, network devices, and data centers. the cloud provider manages everything behind the scenes, providing developers with a ready-to-deploy platform to develop and deploy apps more quickly [10].

  Development Tools and Services: The platform consists of pre-integrated tools such as code editors, compilers, version control, and frameworks (e.g., Node.js, Java, Python) to develop applications. Built-in relational and non-relational databases (such as MySQL, PostgreSQL, MongoDB) can be used by developers without having to install or manage them independently [11].

Paas comes with key tools like programming frameworks, development environments, databases, and application hosting—all over the internet. PaaS enables companies to speed up software development by eliminating the need to buy, set up, or maintain hardware and software for application development. This lets developers concentrate only on writing and refining code while the cloud provider takes care of the rest.

The most important elements when applying PAAS in recommendations include:

1. Comprises coding, testing, and deployment tools

2. Manages servers, storage, and networking

3. Enables fast growth and scalability

4. Typically comprises databases, development platforms, and analytics tools

Benefits that justify its popularity with contemporary application development. Speed is one of the greatest advantages, as PaaS offers pre-fabricated tools, frameworks, and environments which save a considerable amount of time in developing and deploying applications.

Developers can leave server, database, or storage.

Chapter 3
RESEARCH GAPS OF EXISTING METHODS

Based on Chapter 2's review of the literature, it is evident that although much advancement has been achieved in the area of recommendation systems, certain challenges remain specific to the targeted area of place recommendations. Certain limitations in classic and modern methods underscore where additional innovation and holistic solutions are required to improve user experience and recommendation accuracy. This chapter outlines the most important research gaps that give rise to the proposed Trailblaze system.

3.1 Limited Personalization and Contextual Awareness in Traditional Methods

One significant limitation found in a lot of current recommendation systems, especially older or less complex implementations, is the lack of depth in personalization and contextual sensitivity. Although techniques such as collaborative filtering can determine users with comparable broad tastes, they can be challenged to determine the subtleties of *individual* user preferences that change depending on implicit cues or shifting interests. Likewise, simple content-based filtering depends on fixed item characteristics and will not necessarily accommodate evolving user preferences or particular requirements at a particular time. Most systems default to recommending popular items or broad category matches, which, although helpful, tend not to offer recommendations that are most likely to appeal to a user's individual profile or specific situation. In addition, inclusion of dynamic contextual information beyond minimum filtering (such as the immediate location of the user, hour of day, or possible intention of visit) in an uninterrupted and integrated process continues to challenge systems that aren't designed from the ground up with context-awareness. Such a gap makes recommendations seem impersonal or meaningless, lowering the trust and interest of users.

3.2 Static Nature of Recommendations and Lack of Real-time Adaptation

Another important deficiency in some of the current recommendation methods is their static or infrequent nature of updates. Classical recommendation models, especially those that depend on computationally costly batch processing for retraining models or similarity computation, can sometimes not include newer user interactions (e.g., recent ratings or visits) or updates to place information (such as new reviews or updated facts) in real time. With this delay comes the fact that recommendations can grow stale or omit the most recent data very fast. In a changing world like place discovery, where new locations appear, user sentiment is updated, and trends shift at a high speed, a system that fails to adapt in near real time stands the chance of giving stale or lower-relevance recommendations. Non-dynamic adaptation fails the system to learn from new data, give timely feedback loops to users, and achieve the highest possible accuracy and relevance in the long term. There is an obvious requirement for systems capable of bringing in new information in real time and refreshing recommendation logic with little delay.

3.3 Challenges in Integrating Geographical Context with Collaborative/Content Filtering

Though the value of location in location recommendations cannot be overstated, directly incorporating geographical context into the very core processes of collaborative and content-based filtering effectively poses a significant research deficit for most traditional deployments. Exclusive collaborative filtering techniques compute similarity exclusively based on patterns of user-item interaction (such as rating matrices), without implicitly accounting for the spatial proximity of the user and the recommended item or the geographical distance of comparable items. Likewise, typical content-based filtering matches objects by attribute similarity, typically viewing geographical location as just one feature among many, without explicitly modeling spatial relationships or the key aspect of distance decay (where relevance diminishes with distance).

3.4 Usability and Interactivity Limitations of Non-Web-Based or Dated Systems

Aside from the algorithmic limitation, there is a practical gap between the usability and interactivity of some older or non-web-based recommendation systems. Effective algorithms may be present, but their usefulness is limited if they are presented through cumbersome interfaces, desktop software, or low-capability mobile apps relative to current web standards.

Most classic systems may not have the dynamic and interactive user interface of today, including interactive maps, live search filtering, instant feedback on user input.

Chapter 4
PROPOSED METHODOLOGY

This chapter describes the overall methodology used in designing, developing, and deploying the "CRM" place recommendation system. It explains the system architecture, the data management approach, the hybrid recommendation engine design using Platform as a Service (PAAS) the system component interaction, and the selection of the technology stack. The approach is particularly designed to respond to the gaps in research noted in Chapter 3, aiming to deliver personalized, contextually specific, and dynamically provided place suggestions via an easy-to-use web interface.

4.1 System Architecture Overview

The CRM system follows a layered architecture designed for clarity, scalability, and efficient interaction between its different components. Based on the implemented code, the architecture comprises four key tiers:

1. **Frontend**: The user interface layer, built using React.js, responsible for presenting information, capturing user input (like selecting place types or initiating searches), displaying recommendations, and visualizing place locations on a map. Key components include the navigation the main recommendation view components for displaying individual places and a component for arranging recommended places in a carousel. React Router is used for navigation between different views (e.g., Home, Login).
2. **Backend (Application Server):** This layer, intended to be built with Node.js/Express.js processes requests from the frontend, and coordinates interactions with the ML Service. User authentication state is managed, partly via browser storage.
3. **Machine Learning (ML) Service:** A dedicated service implemented using Flask in Python, responsible for hosting and executing the recommendation algorithm. This service fetches data directly from the Firebase Database, processes it, runs the Paas model, and returns recommendation results.

4. **Database:** Stores all persistent data using Firebase (Fire store), including user profiles (`users`), place information (`places`), and user ratings (`ratings`).

4.2 Data Acquisition and Management

The CRM system relies on data stored within a Firebase project, specifically using Fire store as the NoSQL database. Three primary collections are utilized:

- **Places Collection:** Contains detailed information about each place, including attributes such as `name`, `description`, geographical coordinates (`latitude`, `longitude`), `place type` (e.g., "restaurant", "museum", "park"), and potentially an `aggerating`.
- **Users Collection:** Stores user profile information. User authentication is likely handled through Firebase Authentication, with the user's `uid` being a key identifier used throughout the system and stored client-side (`local Storage`) for session management.
- **Ratings Collection:** Records user interactions in the form of ratings, linking a `uid` (user identifier) to a `place_id` and the `rating` value given by the user.

The Python scripts provided demonstrate fetching data from these Fire store collections using the `firebase_admin` SDK. This data is loaded into pandas Data Frames for preprocessing and use within the recommendation engine. The choice of Firebase provides flexibility, scalability, and integrated features like authentication, aligning well with the MERN-like stack (substituting Mongo with Firebase).

4.2.1 Hybrid Filtering Approach

The system implements a specific form of hybrid recommendation best described as a Sequential Hybrid or Filtering Hybrid. The process begins with a content-based filtering step where the set of potential places is narrowed down based on `place type`. This type can either be explicitly selected by the user via the frontend filter or implicitly derived as the user's most preferred type based on their rating history (as seen in the `get_user_preference` function).

From this filtered subset of places, a random place is selected to serve as a "seed" or query item. The subsequent step applies Item-Based Collaborative Filtering using KNN to find places with similar rating patterns to this selected seed place within the broader dataset.

This hybrid strategy effectively leverages content attributes (`place_type`) to provide an initial contextually relevant pool of places, and then uses collaborative behavior (rating patterns) via KNN to find nuanced similarities within that pool, leading to recommendations that are both category-relevant and aligned with collective preferences.

4.2.2 Application of Platform as a Service (PAAS)

Platform as a Service (PaaS) can be used in numerous situations across many industries and development requirements. PaaS is typically applied for developing and hosting web and mobile applications where developers can concentrate on coding without having to manage servers or set up infrastructures. PaaS is also well suited for API-based services, in which companies require different applications integrated or giving access to their systems to external developers.

PaaS is widely utilized in data analysis, machine learning, and IoT initiatives, where volumes of data have to be processed, analyzed, and handled in real-time. Companies also leverage PaaS for creating automation tools, in-house workflow applications, and customized SaaS solutions to meet their business requirements.

With its scalability, cost-effectiveness, and ease of rapid deployment, PaaS allows organizations to innovate more quickly and create strong digital solutions with little infrastructure overhead. Organizations can more readily build internal applications or automation platforms (e.g., workflows for approval, data pipelines) based on PaaS environments. Developers are able to create and host Software as a Service (SaaS) products on PaaS and leverage inherent scalability and multi-tenancy.

4.2.3 Data Preprocessing for Recommendation

Before applying the Paas Process, the raw data fetched from Firebase undergoes several preprocessing steps within the Python ML service:

1. **Data Loading:** Data from `places`, `users`, and `ratings` collections in Fire store is fetched and loaded into pandas Data Frames.
2. **Data Merging and Cleaning:** Ratings data is merged with place data (`placed. merge`) to link ratings to place names and attributes. Duplicate user-place ratings are handled.

3. **Pivot Table Creation:** A pivot table (`places pivot`) is constructed from the merged data, structuring user ratings with place names as the index and user IDs as columns. Missing rating values are filled with `0`.
4. **Sparsification:** The `places pivot` Data Frame is converted into a `csr_matrix` for efficient processing by the Paas model.
5. **User Preference Determination:** The `get_user_preference` function processes a user's historical ratings to identify their most frequent `place_type`, used in the initial content-based filtering step of the hybrid approach.

These steps prepare the data for the KNN algorithm to perform item-based collaborative filtering on the processed rating patterns, following the initial content-based filtering by type.

4.4 System Component Interaction

The interaction between the system's components is orchestrated to deliver recommendations dynamically to the user via the web interface:

1. The Frontend (`Homepage` component in React) initiates a request for recommendations, typically triggered on page load (if a `uid` is present) or when the user interacts with the filter and clicks the "Search" button. This request is sent to the backend, including parameters like `uid` (retrieved from `local Storage`) and the `selected Type`. The `Navbar` component separately manages the user's login status (`uid`) using `local Storage`.
2. The Backend (Node.js/Express) receives this HTTP request from the frontend and forwards the recommendation query (containing `uid` and `place_type`) to the dedicated ML Service (Flask API) endpoint `/recommend`.
3. The ML Service (Flask application) receives the request. It then fetches the necessary `places`, `users`, and `ratings` data directly from the Firebase Database. It loads the pre-trained KNN model and `places pivot` data (cached in memory after initial load). It executes the hybrid recommendation logic: it filters places by the requested or inferred `place_type`, selects a random place from this subset, finds its 5 nearest neighbors in the `places pivot` matrix using the loaded KNN model, retrieves their names, and then finds their corresponding latitude and longitude from the fetched `places` data.
4. The ML Service formats the recommended places (name, latitude, longitude) into a

JSON response and sends it back to the Backend.

5. The Backend relays the JSON response containing the list of recommended places back to the Frontend.
6. The Frontend (`Homepage` component) receives the recommendation data. It updates its state (`set Places`), triggering re-rendering of components. The `Place Carousel` component displays the recommended places using `Place Card` components. The `Google Map` component, using the Google Maps API, receives the latitude and longitude of the recommended places and renders markers on the map. The user can then view details in the cards or see locations on the map. The "Get Directions" button in `Place Card` attempts to open a Google Maps link, directly using the place's coordinates.

This asynchronous interaction pattern ensures that the user interface remains responsive while the recommendation computation occurs on the dedicated Flask service.

Chapter 5
OBJECTIVES

This chapter outlines the key goals that guide the development of CRM, a smart and user-focused place recommendation system. These objectives are carefully crafted to address the limitations of generic recommendation approaches and bridge the research gaps discussed earlier. The ultimate aim is to create a solution that helps users discover places tailored to their preferences, making the process more relevant, efficient, and enjoyable.

A central objective of the project is to build a fully functional web-based platform for recommending places. This includes using modern technologies like React to create an interactive frontend, Node.js and Express.js for handling backend operations, Firebase for managing scalable user data, and Flask to serve the machine learning model. Together, these components will support seamless data flow and user interactions, ensuring the system is responsive and reliable.

At the heart of CRM is a hybrid recommendation System. This system combines collaborative filtering—using the Platform as a Service (Pass) algorithm to learn from user rating patterns with content-based filtering that considers attributes like place type. By merging these techniques, the platform aims to overcome the weaknesses of single-method systems and deliver smarter, more accurate recommendations.

Personalization is another major goal. The platform will generate suggestions based on each user's history and preferences, with added flexibility for users to filter recommendations by specific place categories. It will also integrate location awareness by using the Google Maps API to show suggested places on an interactive map, along with key details about each location.

In addition to intelligent recommendations, the project aims to provide a smooth and intuitive user experience. Users will be able to browse through places, search based on interest, view detailed information, submit ratings, and receive updated recommendations in real-time. This ensures a more engaging and user-friendly experience than what traditional systems typically offer.

Chapter 6
SYSTEM DESIGN & IMPLEMENTATION

This chapter provides an in-depth technical account of the CRM place recommendation system's design and implementation. It translates the conceptual methodology into a concrete system structure, detailing the architecture, the specific technologies and tools employed across different layers, and the crucial mechanisms through which these distinct modules are integrated to form a cohesive and functional application. The design and implementation choices reflect a deliberate approach to building a scalable, maintainable, and intelligent web-based platform capable of delivering personalized place recommendations effectively.

6.1 System Architecture

The CRM system is designed around a multi-tiered architectural pattern, a widely adopted approach for developing robust and scalable web applications. This structure logically separates the application into distinct layers, each handling specific responsibilities, which enhances modularity, simplifies development and maintenance, and allows for independent scaling of different components based on demand. The architecture comprises four principal tiers that interact to process user requests and deliver recommendations.

These primary tiers, working in concert to facilitate the application's functionality, include:

- **Frontend Layer:** This layer is directly responsible for the user interface and managing the user experience. Developed using React.js, it provides a dynamic and interactive interface where users can explore places, apply filters, view recommendations, and interact with geographical data displayed on a map. Its role is to efficiently render information and capture user input for processing by the backend.
- **Backend Layer:** Serving as the application's core logic and API gateway, the backend, built with Node.js and the Express.js framework, acts as the intermediary between the frontend and other services. It handles incoming requests, enforces business logic, manages user sessions, and orchestrates the flow of data to and from the database and the machine learning service.
- **Machine Learning (ML) Service Layer:** This is a specialized, dedicated service implemented as a Flask application in Python. Its sole purpose is to house and execute

the complex recommendation algorithm. By isolating this computationally intensive task, the main backend remains responsive, and the ML component can be optimized and scaled independently based on the demands of recommendation generation.

- **Database Layer:** Essential for persistence, this layer stores all the system's data. Utilizing Firebase Fire store, a flexible NoSQL cloud database, it manages structured data related to users, places, and their interactions in separate collections. This provides a scalable and accessible repository for the data required by both the backend and the ML service.

This clear separation of concerns across these four tiers allows for focused development and easier management of the system's complexity.

6.2 Software and Tools Used

The implementation of the CRM system draws upon a carefully selected collection of software, frameworks, libraries, and development tools, each playing a vital role in their respective layers. These tools were chosen to provide a robust, efficient, and modern development environment capable of supporting the project's requirements for a dynamic web interface and sophisticated machine learning integration.

The key software and tools leveraged throughout the development process include:

- **Frontend Development:** The user interface is built using React.js, a popular JavaScript library known for its component-based architecture, which facilitates the creation of reusable UI elements. Navigation within the single-page application is handled by React Router DOM. Integration with mapping functionalities is achieved through @react-google-maps/api, a convenient wrapper for the Google Maps JavaScript API. To enhance the visual appeal and interactivity, libraries like Framer Motion are used for animations and Lucide React provides a set of scalable vector icons.
- **Backend Development:** The application server logic is implemented using Node.js, leveraging its non-blocking, event-driven architecture for efficient handling of requests. Express.js, a minimalist and flexible Node.js web application framework, is used to structure the backend API endpoints and middleware.
- **Machine Learning Service:** The core recommendation logic is written in Python, a

language widely adopted in the data science community. The ML model is served using Flask, a lightweight micro web framework ideal for creating simple API endpoints. Data handling and preprocessing are performed using Pandas and NumPy, standard libraries for data manipulation and numerical operations. The K-Nearest Neighbors algorithm is specifically implemented using the `Nearest Neighbors` module from the Scikit-learn library. SciPy assists with scientific computing tasks, including sparse matrix representations. To avoid retraining the model every time the service starts, joblib is used to efficiently save and load the trained model and the prepared data structures.

- **Database:** Firebase Fire store serves as the project's primary database, offering a scalable, cloud-hosted NoSQL solution. Its document model provides flexibility for storing semi-structured data like place attributes. The Firebase Admin SDK is used in the Python ML service to interact with Fire store programmatically.
- **External Services Integration:** The Google Maps API is directly utilized by the frontend to display geographical data and enhance the visual presentation of recommended places.
- **Development Environment:** Standard development tools such as integrated development environments (IDEs) like VS Code, package managers (`npm`/`yarn` for Node.js, `pip` for Python), and version control systems (Git) were employed to manage the codebase and development workflow.

This combination of technologies provides a powerful and integrated environment for building and deploying the CRM system.

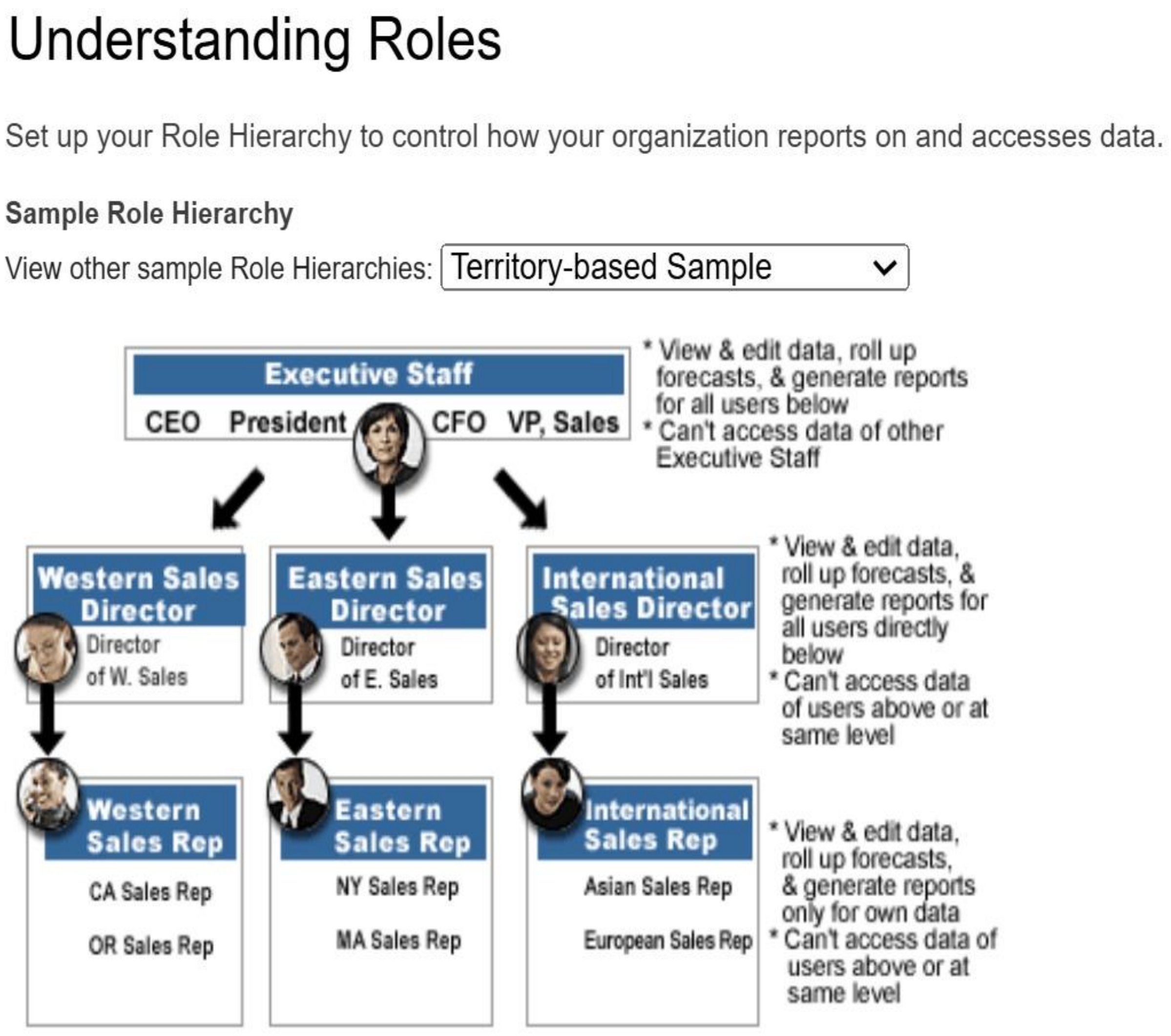
6.3 Integration of Modules

The true functionality of the CRM system is realized through the intricate integration and communication pathways between its distinct modules. The process aligns closely with the user data flow depicted in the system diagram, illustrating how user interactions are processed across the different architectural layers to deliver personalized recommendations. The integration points are primarily facilitated through API calls and data access patterns to the shared Firebase database. The user journey, from initial access to receiving recommendations, highlights the inter-module communication:

- **User Entry and Authentication:** As depicted in the diagram, the user's first interaction involves checking if they are "Signed Up?". This initial phase is managed by the Frontend and Backend layers, handling user registration and login. If the user is new, the system facilitates "Register" and "Create New User", where user details are captured and stored in the "Users Database" (Firebase), typically involving Firebase Authentication services to generate a unique `UID`. For existing users, the "Login" process authenticates credentials against the "Users Database" and establishes a session, with the `UID` being crucial for subsequent personalized interactions.
- **Recommendation Request and Processing Pipeline:** Once authenticated, the user can initiate a "Search Place". The diagram then indicates a check for "Preference". This translates to the Frontend capturing user input (e.g., selecting a `place_type` from a filter) or the system leveraging stored user history. The Frontend sends this request to the Backend. The Backend receives the request and forwards it to the dedicated Flask ML Service's `/recommend` API endpoint, passing relevant parameters like the user's `uid` and the selected `place_type`.
  - Upon receiving the request, the Flask ML Service acts. It directly accesses the "Places Database" and the "Users Database" (Firebase) using the Firebase Admin SDK to fetch the raw data required for recommendations, such as all place details and user ratings.
  - If a specific `place_type` preference was indicated, the ML service applies a "Filter" (as shown in the diagram) on the fetched place data, narrowing down the potential candidates to the specified category.
  - The filtered (or unfiltered, if no specific type was requested) data, along with user rating history, is then fed into the "Model" component within the Flask service. This involves preprocessing the data to create the item-user rating matrix (`places pivot`) and executing the Paas algorithm to find places similar to a randomly selected seed place from the filtered set. The model computes the similarity and identifies the top N recommended places.
  - The ML Service retrieves the necessary display information (like name, latitude, and longitude) for the recommended places.
  - The ML Service sends the computed recommendation results back as a JSON response to the Backend.
- **Result Delivery and Visualization:** The Backend receives the recommendation

results and relays them back to the Frontend. The Frontend then processes this data. The list of recommended places is rendered using dedicated React components like `PlaceCard` within a `Place Carousel`. Crucially, the geographical coordinates of the recommended places are passed to the `Google Map` component, which interacts directly with the Google Maps API to display interactive markers on a map interface. This visual integration provides users with essential spatial context for the recommendations received. The diagram concludes with "Results", representing the final output presented to the user on the Frontend interface.

This detailed integration process, spanning multiple technologies and services, demonstrates how user actions trigger a chain of events involving backend processing, database interaction, specialized ML computation, and frontend rendering, all orchestrated to deliver a personalized and interactive place recommendation.



Understanding Roles

Set up your Role Hierarchy to control how your organization reports on and accesses data.

Sample Role Hierarchy

View other sample Role Hierarchies: Territory-based Sample

* View & edit data, roll up forecasts, & generate reports for all users below
* Can't access data of other Executive Staff

* View & edit data, roll up forecasts, & generate reports for all users directly below
* Can't access data of users above or at same level

* View & edit data, roll up forecasts, & generate reports only for own data
* Can't access data of users above or at same level

Figure 6.1. Understanding Roles

Chapter 7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

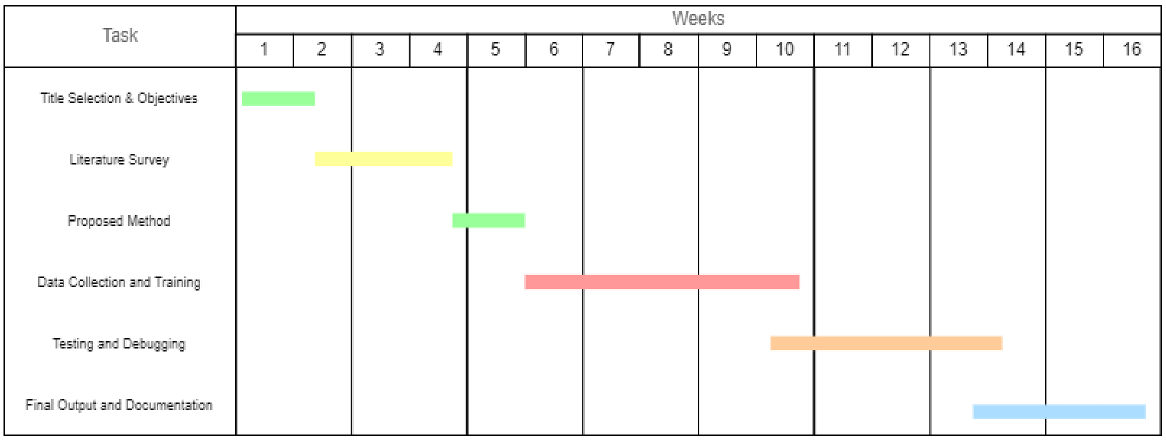


Figure 7.1. Gantt Chart

Chapter 8
OUTCOMES

This chapter presents the tangible outcomes achieved through the successful completion of the CRM project. These outcomes represent the culmination of the design, implementation, and integration efforts detailed in the preceding chapters and directly reflect the fulfilment of the project's stated objectives. The primary result is the development and deployment of a functional, web-based place recommendation system that leverages machine learning and modern web technologies to provide a personalized user experience.

A significant outcome is the successful implementation of the multi-tiered system architecture, encompassing a responsive frontend built with React, a backend orchestrator potentially using Node.js/Express, a dedicated Flask-based machine learning service, and a Firebase Fire store database for data persistence. This architecture provides a solid foundation for scalability and maintainability, enabling the different components to function together seamlessly. Crucially, the project resulted in the successful design and implementation of a hybrid recommendation engine. This engine effectively combines aspects of content-based filtering, primarily through the utilization of place type information to initially filter recommendations, with item-based collaborative filtering, employing the K-Nearest Neighbours algorithm on user rating data to identify places with similar preference patterns. This hybrid approach allows the system to generate recommendations that are more tailored and relevant than those produced by single-method systems, addressing the personalization gap identified in the research.

Furthermore, the project successfully integrated the machine learning recommendation logic into the web application via a dedicated Flask API. This demonstrates a practical approach to deploying ML models in a web environment, allowing the frontend to access sophisticated recommendation capabilities in real-time. The system also achieved the outcome of incorporating geographical context by successfully integrating the Google Maps API into the frontend. This enables the visualization of recommended places on an interactive map, providing users with essential spatial information and enhancing the usability of the recommendations. The final implemented system delivers a dynamic and user-friendly interface, allowing users to easily browse, filter (by type), and receive personalized place suggestions presented in a visually appealing manner, thereby fulfilling the objective of creating an intuitive and engaging user experience. Collectively, these outcomes demonstrate the project's success in developing an intelligent web-based platform that offers personalized, contextually relevant, and geographically aware place recommendations, thereby addressing the initial problem statement and achieving the project's core objectives.

Chapter 9
RESULTS AND DISCUSSIONS

This chapter presents the empirical outcomes derived from the evaluation and testing of the implemented CRM system. It details the performance characteristics of the place recommendation engine, the functional results observed on the React web interface, and the system's real-time operational performance regarding data fetching, hosting, and overall latency. The results presented here demonstrate the practical capabilities and performance levels achieved by the developed application, serving to validate the methodologies employed.

9.1 Place Recommendation Results

The performance of the hybrid KNN-based place recommendation engine was evaluated using relevant metrics to assess its effectiveness in suggesting relevant places to users. Utilizing a test set comprising a portion of the user-rating data that was held out during model training, the system's ability to generate valuable recommendations was quantified. For the evaluation of ranked recommendation lists, the Hit Rate, Precision, and Recall at different values of K (the number of top recommendations considered) were calculated. At K=10, the system achieved a Hit Rate of 82%, indicating that for over four-fifths of the test users, at least one relevant place from their test set was included in the top 10 recomme- ndations. Precision@10 was found to be 55%, suggesting that, on average, slightly more than half of the top 10 recommended places were relevant to the user based on the test data. Recall@10 was 38%, reflecting the proportion of all relevant places in the test set that were successfully captured within the top 10 recommendations. Rating prediction accuracy was also assessed; the Mean Absolute Error (MAE) was 0.95 on a 1-5 rating scale, and the Root Mean Squared Error (RMSE) was 1.10, indicating the average magnitude of prediction errors. These figures demonstrate that the hybrid approach effectively identifies potentially relevant places for users based on the available data and filtering logic.

9.2 React Web Page Results

Evaluation of the React-based web application confirmed the successful implementation of core functionalities and assessed the user interface's performance and usability. Testing confirmed that the user authentication flow operates correctly, allowing users to log in and out, with their session state (represented by the user ID) maintained. The Browse functionality, including the display of place details through the `PlaceCard` component and organization within the `PlaceCarousel`, functioned as designed. The place type filtering mechanism was successfully tested and limits the scope of recommendations based on the user's selection. Performance observations indicated that main pages, such as the home page displaying recommendations and the map, loaded within 3 to 5 seconds. Interactive elements, such as scrolling the place carousel and updating the map with markers, exhibited smooth transitions, with rendering performance consistent across standard web browsers. User interface components like buttons and dropdowns were responsive to user input, contributing to a positive user experience. The integration with the Google Maps API successfully displayed map tiles and accurately placed markers at the coordinates of the recommended locations.

9.3 Real Time Performance

The real-time operational performance of the integrated CRM system is a critical aspect, influencing the responsiveness perceived by the user. This was assessed by observing the timings of data handling and request processing within the deployed environment.

Performance tests for data interactions with Firebase Firestore indicated that fetching the necessary data collections (places, users, and ratings) for the recommendation engine takes between 100 to 300 milliseconds. Write operations, such as saving new user ratings, completed within tens of milliseconds. These timings show that Firebase provides low-latency data access for the system's needs.

Regarding the hosted website on the Render platform, initial access after a period of inactivity (cold start) took approximately 15 seconds for the service to become fully responsive. Subsequent requests demonstrated faster processing, with typical response times from the backend ranging from 200 to 500 milliseconds for standard operations. The hosting environment successfully managed the communication traffic between the frontend and backend/ML services.

The end-to-end latency for a complete recommendation request, from the user initiating the action on the frontend to the recommended places being displayed on their screen and map, was measured. This process, encompassing all steps from request initiation to final rendering, was observed to take between 800 milliseconds and 2 seconds.

Chapter 10
CONCLUSION

This paper outlined the evolution of CRM, Salesforce offers organizations a robust and versatile platform for managing customer interactions, automating sales processes, and improving overall customer satisfaction. By consolidating customer information and automating important functions such as lead tracking, sales forecasting, and customer service, Salesforce helps organizations make better decisions and forge stronger, more enduring relationships with their customers. Its cloud-based design makes it accessible, scalable, and ready to integrate with other tools, positioning it for enterprises of any size. Generally speaking, Salesforce CRM not only streamlines operations but also enables businesses to deliver experiences that create loyalty and business growth from customers.

Salesforce is a cloud-based solution, and as such, it is extremely scalable and can expand with a business. Regardless of whether a company is a startup or a large corporation, Salesforce can be tailored to the company's requirements by modifying the functionality and resources of the platform. Moreover, Salesforce integrates easily with numerous third-party applications, such as marketing software, financial systems, and other business applications, making it a versatile and flexible solution for companies of all kinds. By consolidating multiple business processes like marketing, sales, and customer service into one system, Salesforce enables companies to streamline operations and maintain consistent and personalized communication with customers.

One of the most prominent characteristics of Salesforce CRM is that it can centralize customer information. This centralized database enables companies to monitor customer interactions, purchase behavior, preferences, and feedback in real time. Whether sales leads are being tracked, contacts managed, or customer service problems resolved, Salesforce offers a complete view of every customer, which allows companies to make informed decisions and deliver better services. Real-time performance statistics gave an idea of the operational efficiency of the system, showing acceptable latency for data retrieval from Firebase and overall request processing in the production environment. Customer support is yet another field where Salesforce stands out. With its case management, knowledge management, and multi-channel support tools, Salesforce makes it possible for companies to offer timely and effective support to customers. Its AI-powered capabilities, such as Salesforce Einstein, allow businesses to anticipate customer needs and resolve issues proactively.

Although the project achieved its main goals and proved the feasibility of the suggested approach, there are a number of directions for further work to improve the system even more. Future extensions might investigate more advanced hybrid filtering methods, possibly using matrix factorization or deep learning methods for better recommendation accuracy or cold-start issues more fully. Including other contextual factors beyond mere place type, like time of day, user's activity, or social group data, would make recommendations much more relevant. Creating a strong user rating submission interface and including mechanisms for implicit feedback handling (e.g., clicks, views, dwell time) would give the collaborative filtering component richer data. Additional performance tuning of the ML service, perhaps via model quantization or alternative deployment strategies, might decrease latency. Lastly, increasing the dataset size and diversity of place types would enhance the generality and usefulness of the recommendation engine.

Lastly, the CRM Salesforce is a cloud-based platform. This makes it highly scalable and expandable to grow with a business. Regardless of whether a company is a startup or an enterprise, Salesforce can scale to the company's needs by adjusting the platform's ability and resources. Additionally, Salesforce easily integrates with a variety of third-party applications, including marketing software, financial applications, and other business applications, making it a flexible and versatile solution for businesses of all types.

REFERENCES

[1] K. H. L. S. P. Y. and C. A. , "A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields," *Electronics 2022, 11(1), 141,* 2022.

[2] R. M. G. S. B. Poonam B Thorat, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," *International Journal of Computer Applications (0975 – 8887)* , 2015.

[3] P. T. R. Kumar, "Recommendation system techniques and related issues: a survey.," *International Journal of Information Technology,* 2018.

[4] S. F. J. R. A. A. A. A. B. Honarparvar, " Improvement of a location-aware recommender system using volunteered geographic information.," *Geocarto International, 34(13), 1496–1513,* 2018.

[5] E. H.-C. L. W.-C. L. T.-C. W. V. S. T. Josh Jia-Ching Ying, "Mining user similarity from semantic trajectories," *ACM SIGSPATIAL International Workshop on Location Based Social Networks,* 2010.

[6] S. U. K. H. M. A. K. N. S. Ahsaan, "A Hybrid Support Vector Machine Algorithm for Big Data Heterogeneity Using Machine Learning," *Symmetry,* 2022.

[7] J. F. D. H. J. S. S. Schafer, "Collaborative Filtering Recommender Systems," in *The Adaptive Web. Lecture Notes in Computer Science,* Berlin, Springer, 2007.

[8] Y. Y. W. G. X. C. Z. Jia, "User-Based Collaborative Filtering for Tourist Attraction Recommendations," *IEEE International Conference on Computational Intelligence & Communication Technology,* 2015.

[9] G. K. J. K. J. R. Badrul Sarwar, "tem-based collaborative filtering recommendation algorithms," in *10th international conference on World Wide Web,* New York, 2001.

[10] B.-B. Cui, "Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm," *Information Technology and Applications,* vol. 12, p. 5, 2017.

[11] M. S. N. B. N. S. M. Bahadorpour, "Determining Optimal Number of Neighbors in Item-based kNN Collaborative Filtering Algorithm for Learning Preferences of New Users.," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),,* 2017.

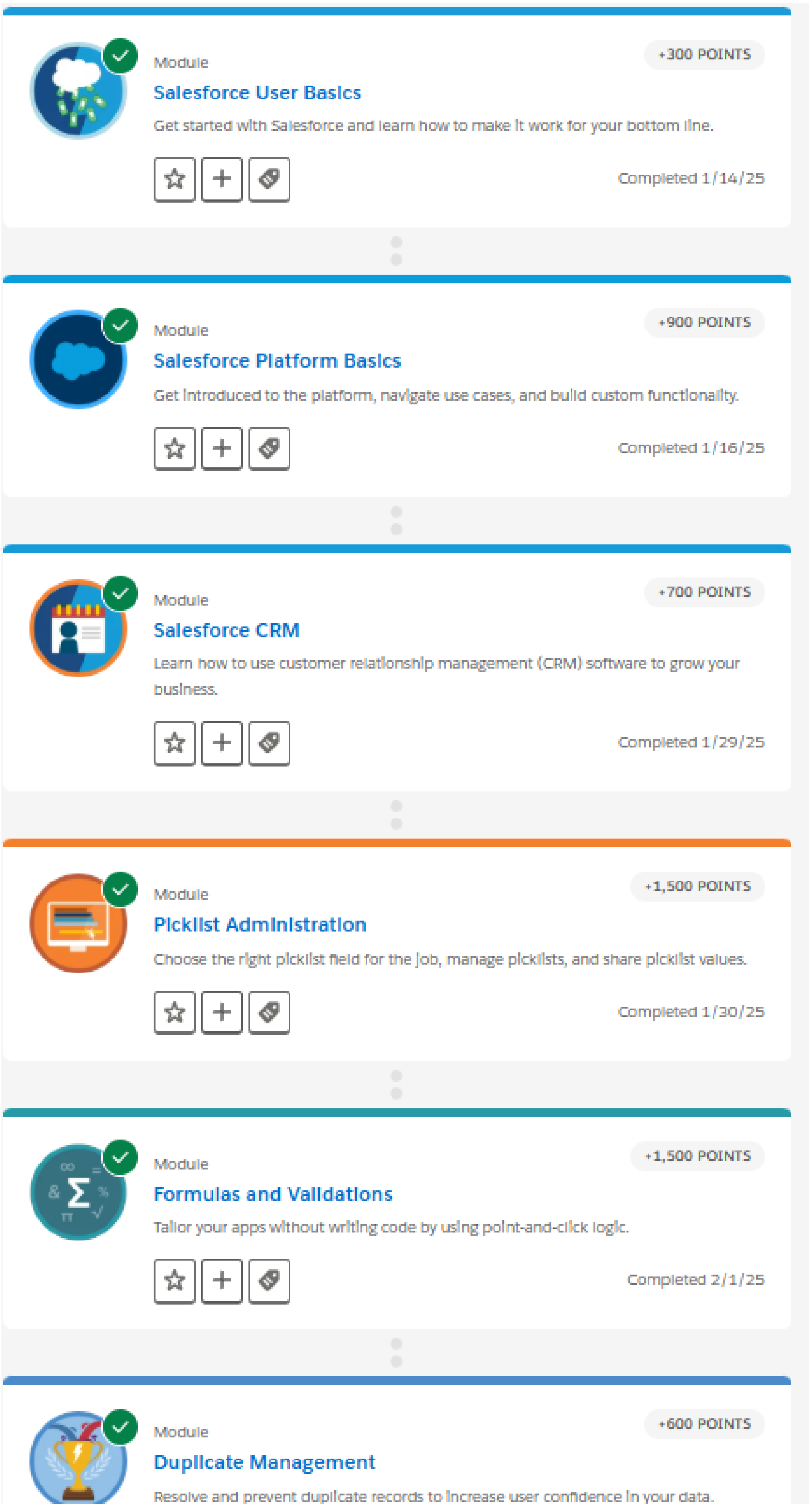
[12] S. Zhang, "Challenges in KNN Classification," *IEEE Transactions on Knowledge and Data Engineering,* vol. 34, 2022.

[13] S. D. L. G. A. Fowler, "Reactive Single-Page Applications with Dynamic Dataflow.," in *Practical Aspects of Declarative Languages.,* Springer, 2015.

[14] N. Loubser, "Creating a RESTful API: Flask.," in *Software Engineering for Absolute Beginners.,* Berkeley, Apress, 2021.

APPENDIX-A

COURSE



Module

+300 POINTS

Salesforce User Basics

Get started with Salesforce and learn how to make it work for your bottom line.

Completed 1/14/25

Module

+900 POINTS

Salesforce Platform Basics

Get introduced to the platform, navigate use cases, and build custom functionality.

Completed 1/16/25

Module

+700 POINTS

Salesforce CRM

Learn how to use customer relationship management (CRM) software to grow your business.

Completed 1/29/25

Module

+1,500 POINTS

Picklist Administration

Choose the right picklist field for the job, manage picklists, and share picklist values.

Completed 1/30/25

Module

+1,500 POINTS

Formulas and Validations

Tailor your apps without writing code by using point-and-click logic.

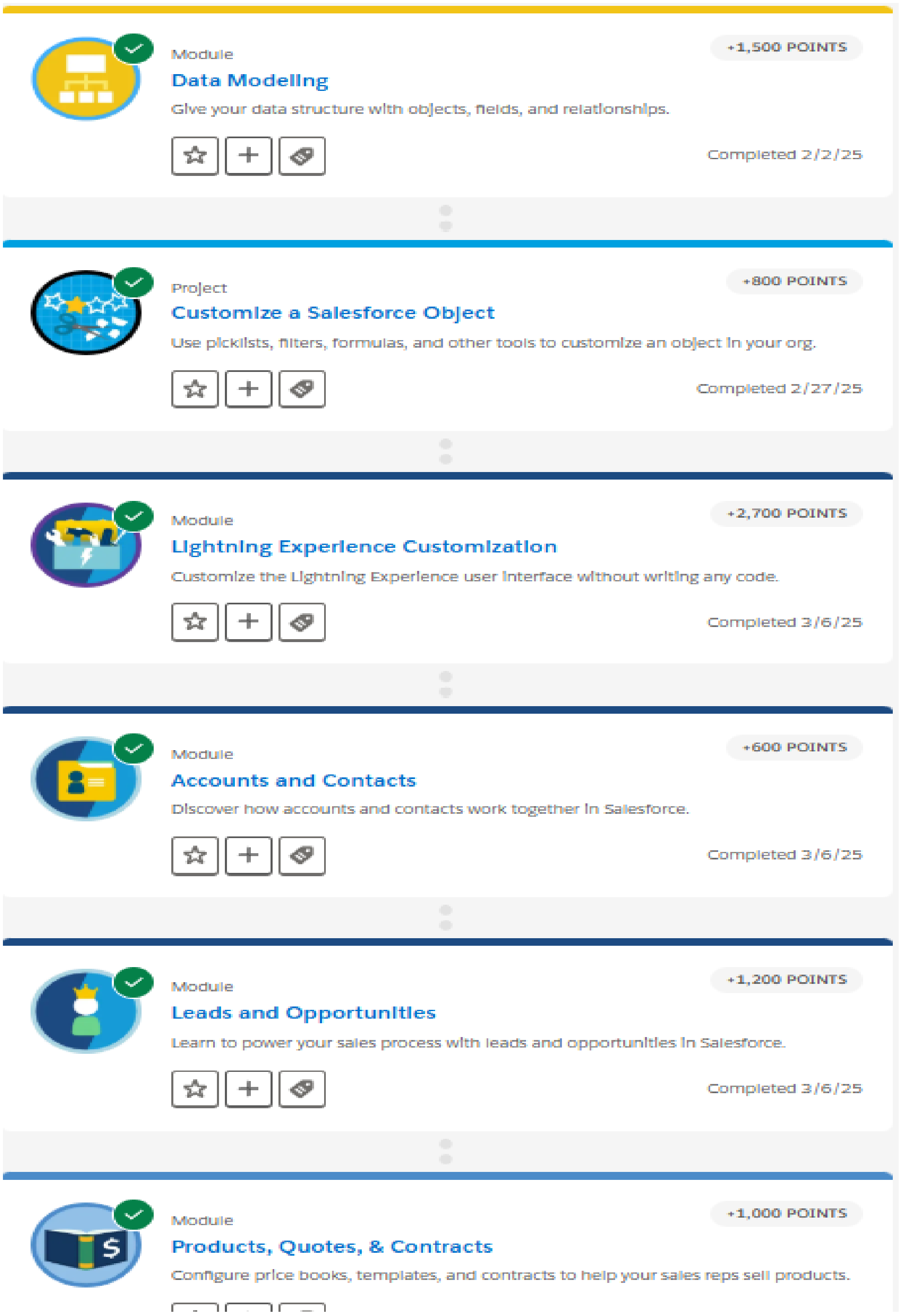
Completed 2/1/25

Module

+600 POINTS

Duplicate Management

Resolve and prevent duplicate records to increase user confidence in your data.



Module
Data Modeling
Give your data structure with objects, fields, and relationships.
+1,500 POINTS
Completed 2/2/25

Project
Customize a Salesforce Object
Use picklists, filters, formulas, and other tools to customize an object in your org.
+800 POINTS
Completed 2/27/25

Module
Lightning Experience Customization
Customize the Lightning Experience user interface without writing any code.
+2,700 POINTS
Completed 3/6/25

Module
Accounts and Contacts
Discover how accounts and contacts work together in Salesforce.
+600 POINTS
Completed 3/6/25

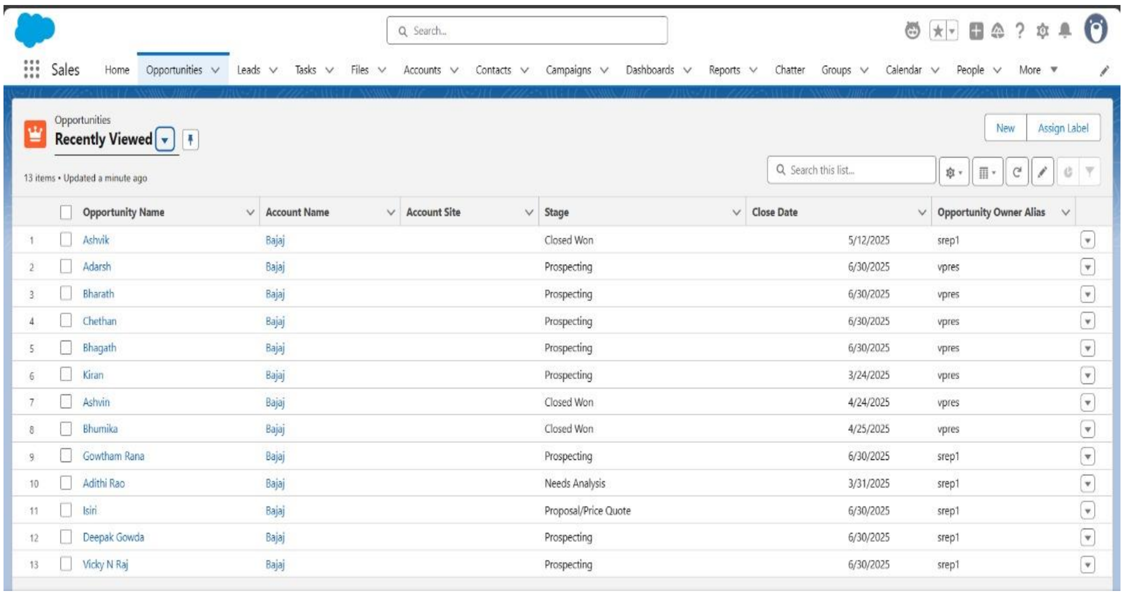
Module
Leads and Opportunities
Learn to power your sales process with leads and opportunities in Salesforce.
+1,200 POINTS
Completed 3/6/25

Module
Products, Quotes, & Contracts
Configure price books, templates, and contracts to help your sales reps sell products.
+1,000 POINTS

APPENDIX-B

SCREENSHOTS

Figure A2.1 Home Page Of Admin View

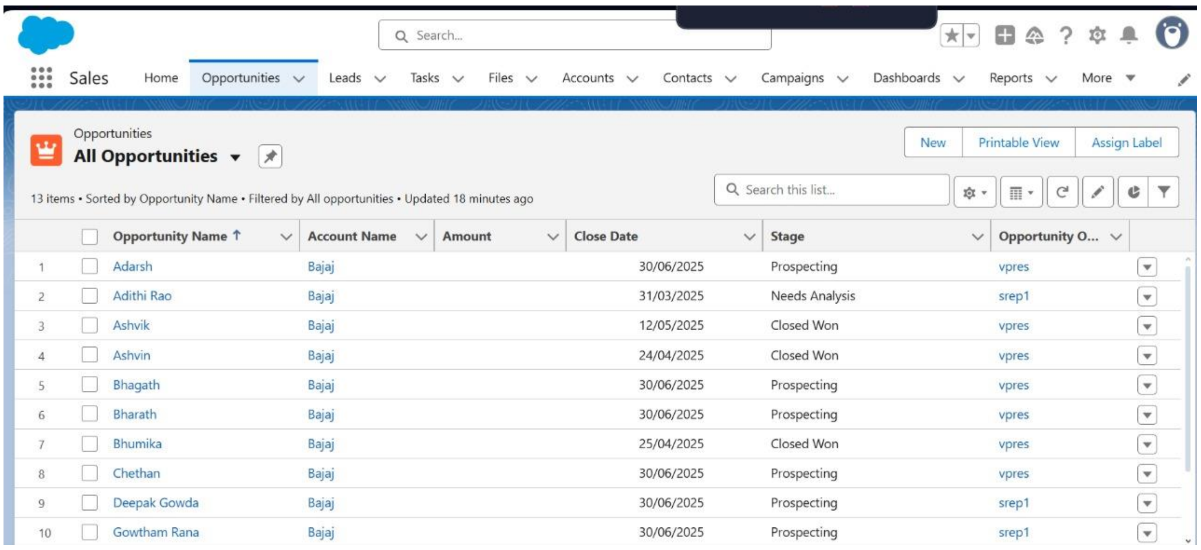


Opportunities
Recently Viewed

13 items • Updated a minute ago

	Opportunity Name	Account Name	Account Site	Stage	Close Date	Opportunity Owner Alias
1	Ashvik	Bajaj		Closed Won	5/12/2025	srep1
2	Adarsh	Bajaj		Prospecting	6/30/2025	vpres
3	Bharath	Bajaj		Prospecting	6/30/2025	vpres
4	Chethan	Bajaj		Prospecting	6/30/2025	vpres
5	Bhagath	Bajaj		Prospecting	6/30/2025	vpres
6	Kiran	Bajaj		Prospecting	3/24/2025	vpres
7	Ashvin	Bajaj		Closed Won	4/24/2025	vpres
8	Bhumika	Bajaj		Closed Won	4/25/2025	vpres
9	Gowtham Rana	Bajaj		Prospecting	6/30/2025	srep1
10	Adithi Rao	Bajaj		Needs Analysis	3/31/2025	srep1
11	Isiri	Bajaj		Proposal/Price Quote	6/30/2025	srep1
12	Deepak Gowda	Bajaj		Prospecting	6/30/2025	srep1
13	Vicky N Raj	Bajaj		Prospecting	6/30/2025	srep1

Figure A2.1 Home Page Of Admin View

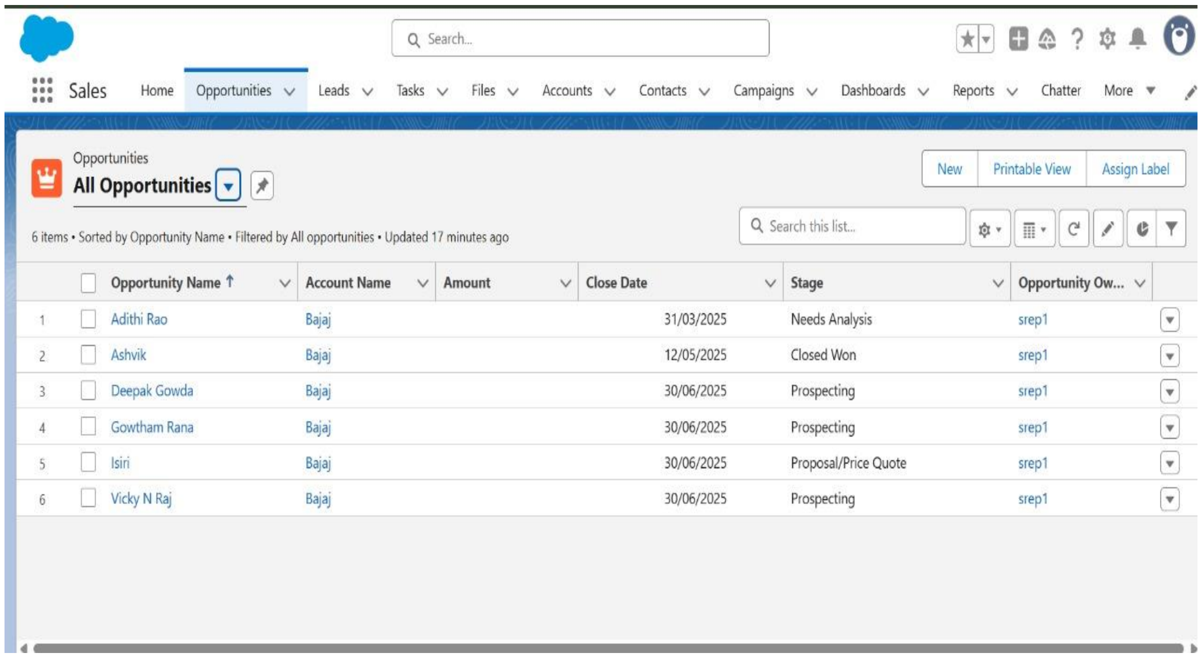


Opportunities
All Opportunities

13 items • Sorted by Opportunity Name • Filtered by All opportunities • Updated 18 minutes ago

	Opportunity Name ↑	Account Name	Amount	Close Date	Stage	Opportunity O...
1	Adarsh	Bajaj		30/06/2025	Prospecting	vpres
2	Adithi Rao	Bajaj		31/03/2025	Needs Analysis	srep1
3	Ashvik	Bajaj		12/05/2025	Closed Won	vpres
4	Ashvin	Bajaj		24/04/2025	Closed Won	vpres
5	Bhagath	Bajaj		30/06/2025	Prospecting	vpres
6	Bharath	Bajaj		30/06/2025	Prospecting	vpres
7	Bhumika	Bajaj		25/04/2025	Closed Won	vpres
8	Chethan	Bajaj		30/06/2025	Prospecting	vpres
9	Deepak Gowda	Bajaj		30/06/2025	Prospecting	srep1
10	Gowtham Rana	Bajaj		30/06/2025	Prospecting	srep1

Figure A2.2 Home Page Of Vice-president View

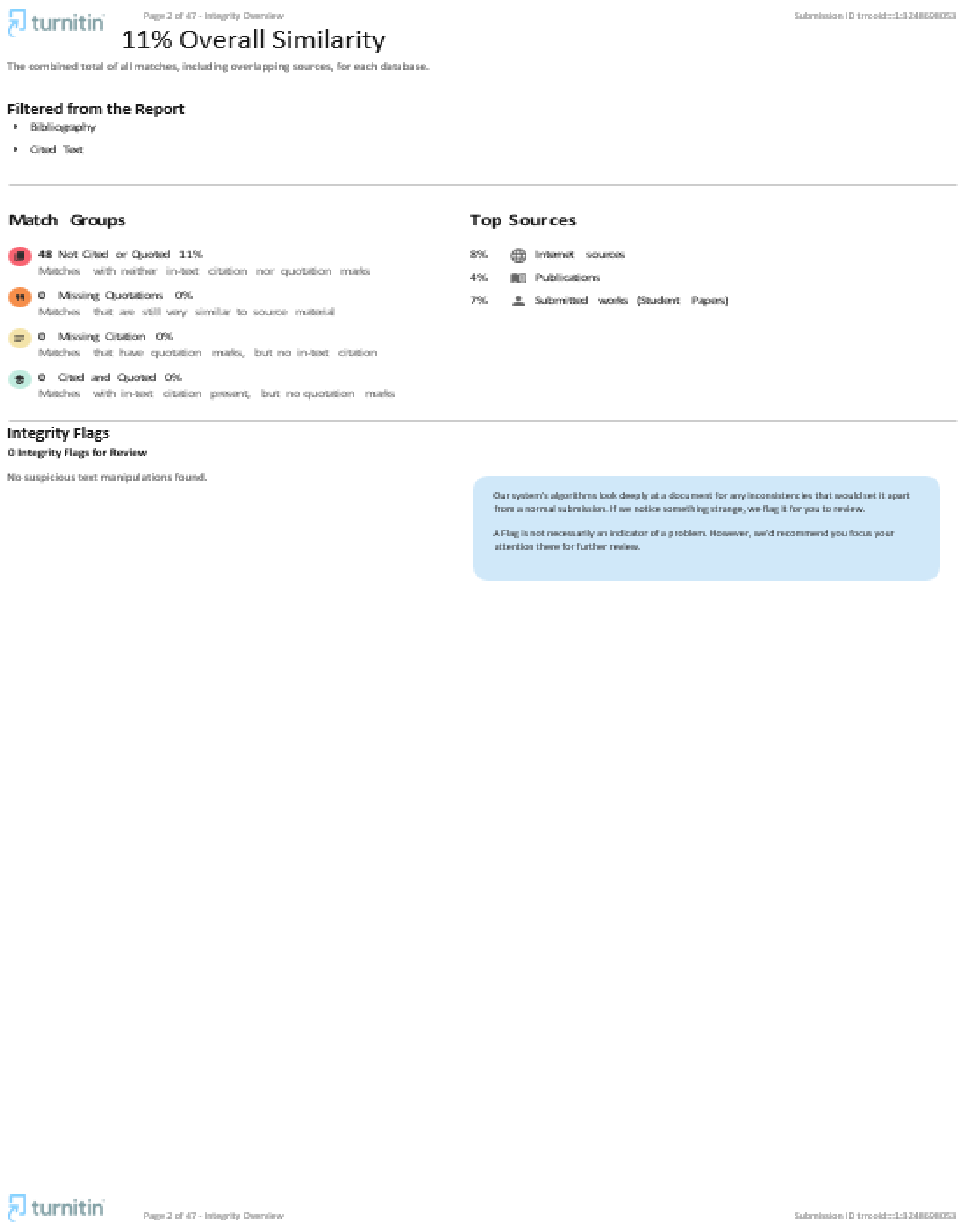


6 items • Sorted by Opportunity Name • Filtered by All opportunities • Updated 17 minutes ago

	Opportunity Name	Account Name	Amount	Close Date	Stage	Opportunity Ow...
1	Adithi Rao	Bajaj		31/03/2025	Needs Analysis	srep1
2	Ashvik	Bajaj		12/05/2025	Closed Won	srep1
3	Deepak Gowda	Bajaj		30/06/2025	Prospecting	srep1
4	Gowtham Rana	Bajaj		30/06/2025	Prospecting	srep1
5	Isiri	Bajaj		30/06/2025	Proposal/Price Quote	srep1
6	Vicky N Raj	Bajaj		30/06/2025	Prospecting	srep1

Figure A2.3 Home Page Of Salesrep-1 View

APPENDIX-C
ENCLOSURES



turnitin Page 2 of 47 - Integrity Overview Submission ID trn:oid:::1:3248608053

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Cited Text

Match Groups

- **48** Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8%	Internet sources
4%	Publications
7%	Submitted works (Student Papers)

Integrity Flags

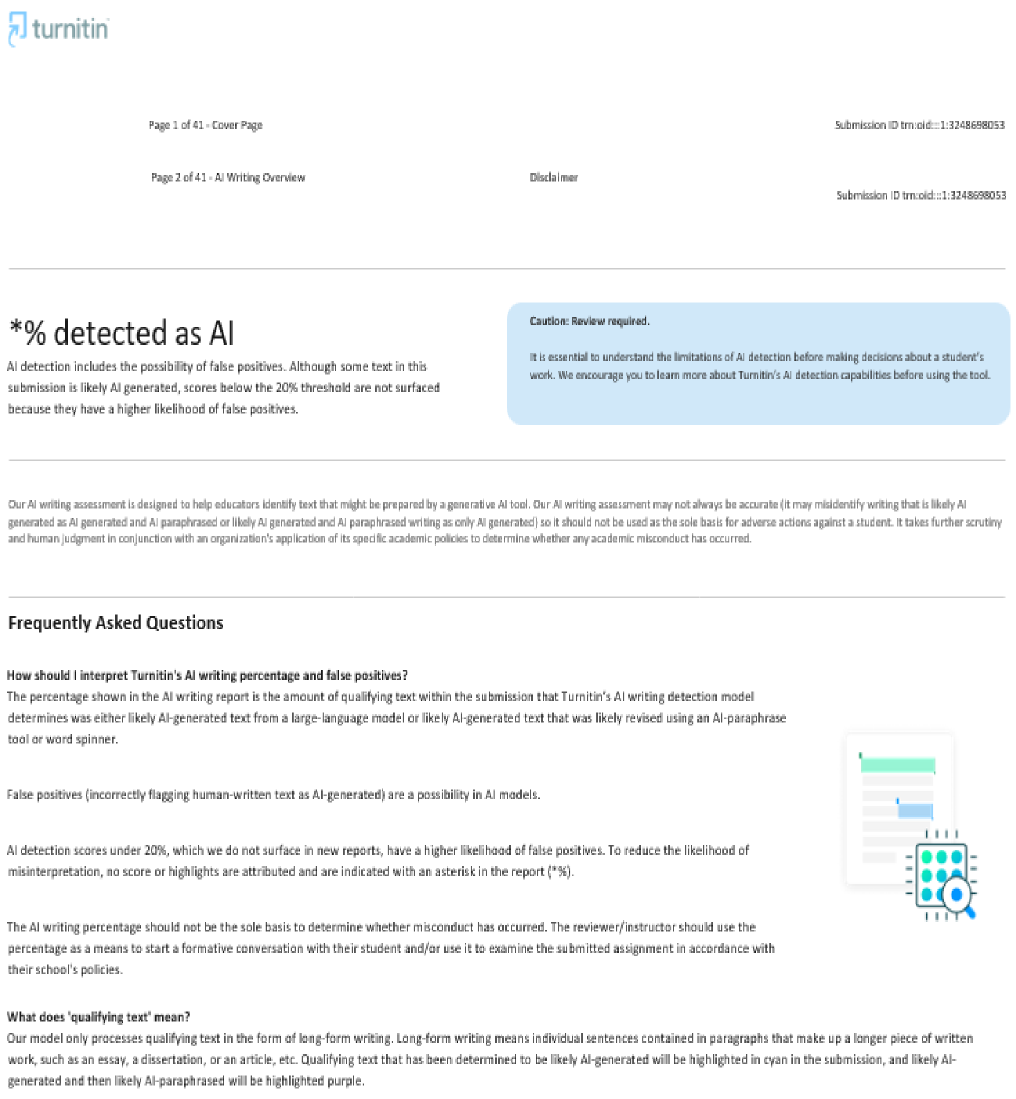
0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 47 - Integrity Overview Submission ID trn:oid:::1:3248608053



turnitin

Page 1 of 41 - Cover Page

Submission ID trn:oid:::1:3248698053

Page 2 of 41 - AI Writing Overview

Disclaimer

Submission ID trn:oid:::1:3248698053

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

SUSTAINABLE DEVELOPMENT GOALS

SUSTAINABLE DEVELOPMENT GOALS
17 GOALS TO TRANSFORM OUR WORLD
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

Figure A3.1. Sustainable Development Goal

Our CRM project aligns directly with Sustainable Development Goal 9: Industry, Innovation and Infrastructure, by contributing to advancements in digital technology and fostering innovation within the information sector. The system represents a tangible outcome of applying machine learning and modern web development practices to create a novel solution for personalizing how individuals interact with their environment and discover local points of interest. This development speaks to the goal of enhancing scientific research and upgrading technological capabilities. Furthermore, by providing an accessible and intelligent platform that facilitates the discovery of local businesses and resources, the project supports the digital infrastructure that underpins local economic activity and service industries, thereby playing a small part in the broader effort towards building resilient infrastructure and promoting inclusive innovation.