IBM Innovation Center for Education

PHEMESOFT Software That Matters

TRUSTED DELIVERY PARTNER for IBM ICE

University of Petroleum and Energy Studies

Internship - Low Level Design
on
Sales and Lead CRM

SUBMITTED BY

Rohan Bakshi (R2142211066)

Lavanya Raj (R2142211045)

Vilish Kumar (R2142211034)

Harsh Chaudhary(R2142211041)

Sidharth Bansal(R2142211073)

Guide By:

Dr.Lalit Kumar

PROJECT TEAM MEMBERS:

S. No.	Name	SAP ID	E-mail	Phone number	Branch
1	Rohan Bakshi	500096302	500096302@stu.upes.ac.in	6203692405	B.TechCSE (Ccvt)Batch 6
2	Lavanya Raj	500096244	500096244@stu.upes.ac.in	8789447556	B.Tech CSE(Ccvt) Batch 6
3	Vilish Kumar	500095673	500095673@stu.upes.ac.in	7983219426	B.TechCSE (Ccvt) Batch 6
4	Harsh Chaudhary	500095565	500095565@stu.upes.ac.in	9149073580	B. Tech CSE (Ccvt) Batch 5
5	Sidharth Bansal	500096346	500096346@stu.upes.ac.in	9457094600	B.Tech CSE(Ccvt) Batch 6

Table of Contents

1.Introduction

This document outlines the framework and guidelines for implementing and managing a Cloud-Based Customer Relationship Management (CRM) system. The purpose is to provide comprehensive information regarding the scope, intended audience, and an overview of the system.

1.1. Scope of the Document

The scope of this document includes the design, development, implementation, and maintenance of the Cloud-Based CRM system. It covers the following aspects:

- System architecture and design
- Functional and non-functional requirements
- User roles and permissions
- Data management and security protocols
- Integration with other systems
- Maintenance and support procedures

1.2. Intended Audience

This document is intended for a wide range of stakeholders, including:

- Project Managers
- System Developers and Engineers
- Business Analysts
- Quality Assurance Teams
- Customer Support Representatives
- End-users of the CRM system
- External partners and vendors involved in the CRM project

1.3. System Overview

The Cloud-Based CRM system is designed to help organizations manage their interactions with current and potential customers using a web-based platform. It provides tools for:

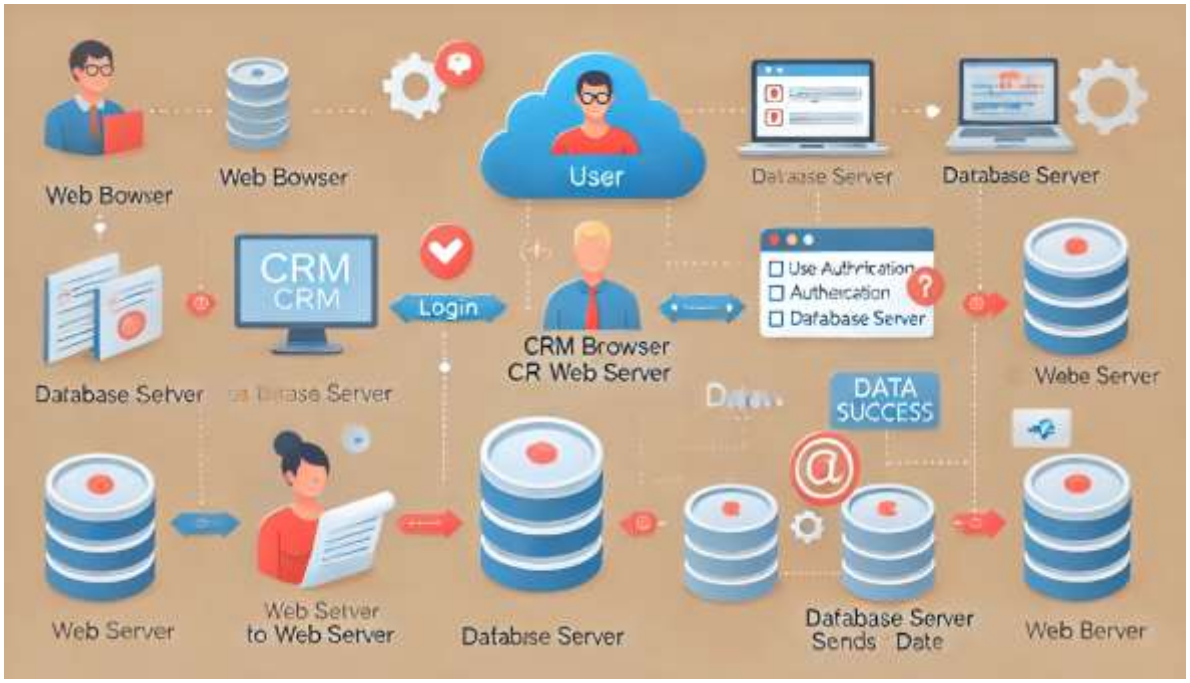
- Tracking customer interactions and history
- Managing customer data and preferences
- Automating marketing and sales processes
- Enhancing customer support and service
- Generating reports and analytics for business insights

The system aims to improve customer satisfaction, streamline business processes, and increase overall efficiency and productivity in managing customer relationships. The cloud-based nature of the system offers additional benefits such as:

- Scalability: Easily scalable to meet the growing needs of the business.
- Accessibility: Accessible from any location with an internet connection.
- Cost-Effectiveness: Reduces the need for physical infrastructure and maintenance.

2. Low Level System Design

2.1 Sequence Diagram



Components:

1.User: The person interacting with the CRM system through a web browser.

2.Web Browser: The application used by the user to access the CRM system. It sends and receives requests and responses to and from the CRM Web Server.

3.CRM Web Server: The server that handles user requests, processes them, and interacts with the database server to fetch or store data.

4.Database Server: The server where the CRM data is stored. It handles queries from the CRM Web Server and returns the requested data.

Interactions:

1.User Logs In:

- **User**: The user initiates the login process by entering their credentials (username and password) in the Web Browser.
- **Web Browser**: The Web Browser sends the login request, including the user's credentials, to the CRM Web Server.

2.Login Request:

- **CRM Web Server**: The CRM Web Server receives the login request from the Web Browser and processes it. It needs to verify the user's credentials against the stored data.

3.User Authentication:

- **CRM Web Server**: To authenticate the user, the CRM Web Server queries the Database Server with the provided credentials.
- **Database Server**: The Database Server receives the query, checks the credentials against its stored user data, and returns the authentication result (success or failure) to the CRM Web Server.

4.Authentication Result:

- **CRM Web Server**: Based on the result from the Database Server, the CRM Web Server determines whether the login is successful or not. It then sends the appropriate response back to the Web Browser.

5.Login Response:

- **Web Browser**: The Web Browser receives the login success response from the CRM Web Server and allows the user to proceed to the next steps within the CRM system.

6.User Initiates Data Retrieval:

- **User**: The authenticated user initiates a request to retrieve certain data (e.g., customer information) via the Web Browser.
- **Web Browser**: The Web Browser sends the data retrieval request to the CRM Web Server.

7.Data Retrieval Request:

- **CRM Web Server**: The CRM Web Server receives the data retrieval request and needs to fetch the requested data from the Database Server.

8.Querying Database:

- **CRM Web Server**: The CRM Web Server queries the Database Server for the specific data requested by the user.
- **Database Server**: The Database Server processes the query, retrieves the requested data, and returns it to the CRM Web Server.

9.Returning Data:

- **CRM Web Server**: The CRM Web Server receives the data from the Database Server and prepares it for the user.
- **CRM Web Server**: The CRM Web Server sends the retrieved data back to the Web Browser.

10.Displaying Data:

- **Web Browser**: The Web Browser receives the data from the CRM Web Server and displays it to the user, completing the data retrieval process.

2.2 Navigation Flow/UI Implementation

The navigation flow of the cloud-based CRM system defines how users move from one interface screen to another. It ensures a smooth and intuitive user experience, enabling users to efficiently access and manage customer information, track interactions, and perform various CRM-related tasks.

1. **Login Page**:
    - **Components**: Username field, password field, login button, forgot password link.
    - **Flow**:
        - User enters username and password.
        - User clicks the login button.
        - Browser sends login credentials to the CRM Web Server for authentication.
        - If authentication is successful, user is redirected to the Dashboard.
2. **Dashboard**:
    - **Components**: Navigation menu, summary widgets (e.g., recent activities, key metrics), quick access buttons (e.g., add new contact, view reports).
    - **Flow**:
        - User views summary information.
        - User can click on navigation items to access different sections like Contacts, Leads, Opportunities, Reports, and Settings.
        - User can click on quick access buttons to perform actions directly from the dashboard.
3. **Contacts Page**:
    - **Components**: Search bar, contact list, add new contact button, contact details panel.
    - **Flow**:

- User searches for a contact using the search bar.
- User selects a contact from the list to view or edit details.
- User can click the add new contact button to open the new contact form.

4. **Contact Details Page**:
   - **Components**: Contact information fields (name, email, phone, address), interaction history, notes section, save button.
   - **Flow**:
     - User views detailed information about a contact.
     - User can update contact information and click the save button to store changes.
     - Interaction history shows past communications with the contact.
     - Notes section allows users to add or edit notes related to the contact.
5. **Leads Page**:
   - **Components**: Search bar, leads list, add new lead button, lead details panel.
   - **Flow**:
     - Similar to Contacts Page, but focused on potential customers (leads).

2.3 Screen Validations, Defaults, and Attributes

Screen validations ensure that user inputs are correct and complete before the data is processed and stored in the system. Default values help streamline data entry by pre-filling fields with common or previously used values. Attributes define the properties and constraints of each data field.

1. **Login Page**:
   - **Validations**:
     - **Username**: Required, must be a valid email format.
     - **Password**: Required, minimum length of 8 characters.
   - **Defaults**: None.
   - **Attributes**:
     - **Username Field**: `type="email"`, `required`, `placeholder="Enter your email"`.
     - **Password Field**: `type="password"`, `required`, `placeholder="Enter your password"`.
2. **Dashboard**:
   - **Validations**: Not applicable (mainly navigation and display page).
   - **Defaults**: Not applicable.
   - **Attributes**:
     - **Widgets**: Dynamic content based on user data.
     - **Navigation Menu**: Links to other sections (Contacts, Leads, etc.).

2.4 Client-Side Validation Implementation

Client-side validation is implemented to enhance user experience by providing immediate feedback on input errors. This section describes the methods and technologies used to perform these validations within the user's browser, such as:

- JavaScript and HTML5 form validation techniques.
- Real-time validation feedback (e.g., highlighting erroneous fields).
- Preventing form submission until all validations are satisfied.

2.5 Server-Side Validation Implementation

Server-side validation ensures data integrity and security by validating inputs on the server. This section covers the techniques and frameworks used to implement these validations and handle errors gracefully, including:

- Ensuring all data received from the client is sanitized.
- Validating data against business rules and database constraints.
- Providing meaningful error messages back to the client for correction.

2.6 Components Design Implementation

This section describes the design and implementation of various system components, including:

- **User Authentication and Authorization:** Mechanisms for user login, registration, password management, and access control.
- **Data Management Modules:** Components for managing customer data, interaction history, and other CRM-related information.
- **Communication and Notification Services:** Tools for managing emails, alerts, and notifications to customers and internal users.
- **Reporting and Analytics Tools:** Modules for generating insights and reports from CRM data.
- **Integration Modules for Third-Party Systems:** Interfaces for connecting with external systems such as marketing automation tools, ERP systems, and social media platforms.

2.7 Configurations/Settings

Configurations and settings provide flexibility and customization options for the CRM system. This section covers:

- **System Configuration Options:** Settings that control system behavior and features.
- **User Preference Settings:** Customizable options for end-users to tailor the CRM experience to their needs.
- **Administrative Controls:** Tools for system administrators to manage users, permissions, and system settings.
- **Environment-Specific Settings:** Configurations for different environments (e.g., development, testing, production) to ensure consistent behavior across all stages of deployment.

2.8 Interfaces to Other Components

The CRM system interacts with various external components and services. This section describes the interfaces and integration points, including:

- **APIs for Integration with Other Software Systems:** RESTful APIs, SOAP services, and other integration mechanisms to connect with third-party applications.
- **Data Import/Export Mechanisms:** Tools for importing data from legacy systems and exporting CRM data for reporting or backup purposes.
- **Communication Protocols:** Standards and protocols for data exchange, such as HTTP/HTTPS, WebSocket, and SMTP.
- **Authentication and Authorization Interfaces for External Systems:** OAuth, SAML, and other mechanisms to ensure secure interactions with external services.

This detailed design documentation ensures a thorough understanding of the Cloud-Based CRM system's architecture and implementation, facilitating effective development, deployment, and maintenance.

3. Data Design

List of Key Schemas/Tables in Database

In a cloud-based CRM (Customer Relationship Management) system, the following key schemas/tables are essential for managing customer data, interactions, and related business processes:

1. **Customer Information Table (Customers)**
   - **Fields:** CustomerID, FirstName, LastName, Email, Phone, Address, DateOfBirth, Gender, AccountManagerID, CreatedDate, ModifiedDate
2. **Contact Information Table (Contacts)**
   - **Fields:** ContactID, CustomerID, FirstName, LastName, Email, Phone, Relationship, CreatedDate, ModifiedDate
3. **Sales/Leads Table (SalesLeads)**
   - **Fields:** LeadID, CustomerID, LeadSource, LeadStatus, PotentialValue, CreatedDate, ModifiedDate
4. **Opportunities Table (Opportunities)**

- **Fields:** OpportunityID, CustomerID, OpportunityName, OpportunityValue, Stage, Probability, CloseDate, CreatedDate, ModifiedDate

5. **Interaction/Activity Table (Activities)**
   - **Fields:** ActivityID, CustomerID, ActivityType, Description, ActivityDate, FollowUpDate, CreatedDate, ModifiedDate
6. **Product/Service Table (ProductsServices)**
   - **Fields:** ProductID, ProductName, Description, Price, Category, CreatedDate, ModifiedDate
7. **Transactions Table (Transactions)**
   - **Fields:** TransactionID, CustomerID, ProductID, TransactionDate, Amount, PaymentMethod, CreatedDate, ModifiedDate
8. **Support Tickets Table (SupportTickets)**
   - **Fields:** TicketID, CustomerID, IssueDescription, Status, Priority, AssignedAgentID, CreatedDate, ModifiedDate
9. **User Table (Users)**
   - **Fields:** UserID, Username, PasswordHash, Role, CreatedDate, ModifiedDate
10. **Access Logs Table (AccessLogs)**
    - **Fields:** LogID, UserID, Action, Timestamp, IPAddress, Details
11. **Account Managers Table (AccountManagers)**
    - **Fields:** AccountManagerID, FirstName, LastName, Email, Phone, CreatedDate, ModifiedDate
12. **Roles and Permissions Table (RolesPermissions)**
    - **Fields:** RoleID, RoleName, PermissionLevel, CreatedDate, ModifiedDate

3.2. Details of Access Levels on Key Tables in Scope

Access levels in a cloud-based CRM system need to be carefully defined to ensure data security and appropriate access control. The following are typical access levels:

1. **Admin Access:**
   - Full read and write access to all tables.
   - Can create, modify, and delete records.
   - Manage user roles and permissions.
2. **Sales Team Access:**
   - Read and write access to Customer Information, Sales/Leads, Opportunities, and Interactions/Activity tables.
   - Read-only access to Product/Service and Transactions tables.
   - No access to User and Access Logs tables.
3. **Support Team Access:**
   - Read and write access to Customer Information, Interactions/Activity, and Support Tickets tables.
   - Read-only access to Sales/Leads and Opportunities tables.
   - No access to User and Access Logs tables.
4. **Customer Access (via portal):**
   - Read-only access to their own Customer Information and Transactions.
   - Can create and view Support Tickets.
   - No access to Sales/Leads, Opportunities, and Interactions/Activity tables.
5. **Manager Access:**
   - Read and write access to all tables within their team's scope.
   - Can view and generate reports from Transactions and Sales/Leads.
   - Read-only access to User table to manage team members.

3.3. Key Design Considerations in Data Design

When designing data for a cloud-based CRM system, several key considerations must be taken into account to ensure efficiency, scalability, and security:

1. **Data Integrity:**

   - Ensure that all data entries are validated and that relationships between tables are maintained (e.g., foreign keys).
2. **Scalability:**
   - Design the database to handle increasing amounts of data and concurrent access without performance degradation.
3. **Security:**
   - Implement encryption for sensitive data both at rest and in transit.
   - Regularly audit and review access controls and permissions.
4. **Performance:**
   - Optimize queries and indexes to ensure quick retrieval of data.
   - Use caching mechanisms where appropriate to reduce load on the database.
5. **Redundancy and Backup:**
   - Implement regular backups and ensure data redundancy to prevent data loss.
6. **Compliance:**
   - Ensure that the database design complies with relevant data protection regulations (e.g., GDPR, CCPA).
7. **Usability:**
   - Design the schema to be intuitive for users and developers, facilitating ease of use and maintenance.
8. **Integration:**
   - Ensure the database can integrate with other systems and services (e.g., marketing automation tools, email services).

4.Details of Access Levels on Key Tables in Scope

Access levels in a cloud-based CRM system must be clearly defined to ensure data security and appropriate access control for different user roles. Below are the typical access levels for various roles within the system:

1. Admin Access

- **Full Access:** Can create, read, update, and delete (CRUD) records in all tables.
- **Tables:** Customers, Contacts, SalesLeads, Opportunities, Activities, ProductsServices, Transactions, SupportTickets, Users, AccessLogs, AccountManagers, RolesPermissions.

2. Sales Team Access

- **Customers:**
  - **Read/Write:** Can view and update customer information.
- **Contacts:**
  - **Read/Write:** Can view and update contact information.
- **SalesLeads:**
  - **Read/Write:** Can create, view, update, and delete sales leads.
- **Opportunities:**
  - **Read/Write:** Can create, view, update, and delete sales opportunities.
- **Activities:**
  - **Read/Write:** Can create, view, update, and delete activities and interactions.
- **ProductsServices:**
  - **Read-Only:** Can view product and service information.
- **Transactions:**
  - **Read-Only:** Can view transaction information.
- **SupportTickets:**
  - **No Access**
- **Users:**
  - **No Access**
- **AccessLogs:**
  - **No Access**
- **AccountManagers:**
  - **Read-Only:** Can view account manager information.

- **RolesPermissions:**
  - **No Access**

3. Support Team Access

- **Customers:**
  - **Read-Only:** Can view customer information.
- **Contacts:**
  - **Read-Only:** Can view contact information.
- **SalesLeads:**
  - **Read-Only:** Can view sales leads.
- **Opportunities:**
  - **Read-Only:** Can view opportunities.
- **Activities:**
  - **Read/Write:** Can create, view, update, and delete activities and interactions.
- **ProductsServices:**
  - **Read-Only:** Can view product and service information.
- **Transactions:**
  - **Read-Only:** Can view transaction information.
- **SupportTickets:**
  - **Read/Write:** Can create, view, update, and delete support tickets.
- **Users:**
  - **No Access**
- **AccessLogs:**
  - **No Access**
- **AccountManagers:**
  - **No Access**
- **RolesPermissions:**
  - **No Access**

4. Customer Access (via Portal)

- **Customers:**
  - **Read-Only:** Can view their own customer information.
- **Contacts:**
  - **Read-Only:** Can view their own contact information.
- **SalesLeads:**
  - **No Access**
- **Opportunities:**
  - **No Access**
- **Activities:**
  - **No Access**
- **ProductsServices:**
  - **Read-Only:** Can view product and service information.
- **Transactions:**
  - **Read-Only:** Can view their own transactions.
- **SupportTickets:**
  - **Read/Write:** Can create and view their own support tickets.
- **Users:**
  - **No Access**
- **AccessLogs:**
  - **No Access**
- **AccountManagers:**
  - **No Access**
- **RolesPermissions:**
  - **No Access**

5. Manager Access

- **Customers:**
  - **Read/Write:** Can view and update customer information within their team's scope.
- **Contacts:**
  - **Read/Write:** Can view and update contact information within their team's scope.
- **SalesLeads:**
  - **Read/Write:** Can create, view, update, and delete sales leads within their team's scope.
- **Opportunities:**
  - **Read/Write:** Can create, view, update, and delete opportunities within their team's scope.
- **Activities:**
  - **Read/Write:** Can create, view, update, and delete activities and interactions within their team's scope.
- **ProductsServices:**
  - **Read-Only:** Can view product and service information.
- **Transactions:**
  - **Read-Only:** Can view transaction information.
- **SupportTickets:**
  - **Read/Write:** Can create, view, update, and delete support tickets within their team's scope.
- **Users:**
  - **Read-Only:** Can view user information to manage team members.
- **AccessLogs:**
  - **Read-Only:** Can view access logs for auditing purposes.
- **AccountManagers:**
  - **Read/Write:** Can view and update account manager information within their team's scope.
- **RolesPermissions:**
  - **No Access**

4.1 Key Design Considerations in Data Design for a Cloud-Based CRM

When designing the data structure for a cloud-based CRM (Customer Relationship Management) system, it's crucial to consider several factors to ensure the system is robust, efficient, and scalable. Below are the key design considerations:

1. Data Integrity

- **Consistency:** Ensure that data across tables is consistent and accurate. Use foreign keys to maintain relationships between tables (e.g., linking customers to their transactions).
- **Validation:** Implement data validation rules to ensure data integrity. For example, email addresses should follow a standard format.

2. Scalability

- **Horizontal Scaling:** Design the system to handle increasing loads by adding more servers. Use partitioning and sharding techniques to distribute data across multiple databases.
- **Vertical Scaling:** Optimize database schemas and indexing to support vertical scaling if needed.

3. Security

- **Data Encryption:** Encrypt sensitive data both at rest and in transit to protect against unauthorized access and data breaches.
- **Access Control:** Implement fine-grained access control to restrict access to data based on user roles and permissions.
- **Audit Trails:** Maintain detailed logs of all access and modification activities for auditing and compliance purposes.

4. Performance

- **Indexing:** Use indexes to speed up query performance on frequently accessed fields.
- **Query Optimization:** Optimize SQL queries to reduce load times and improve response times.

- **Caching:** Implement caching mechanisms to store frequently accessed data in memory, reducing the need for repeated database queries.

5. Redundancy and Backup

- **Data Replication:** Use data replication techniques to ensure high availability and data redundancy. This can help in disaster recovery scenarios.
- **Regular Backups:** Implement automated backup processes to regularly back up data, ensuring data can be restored in case of data loss.

6. Compliance

- **Regulatory Compliance:** Ensure the database design complies with relevant data protection regulations (e.g., GDPR, CCPA). This includes data retention policies, consent management, and the right to be forgotten.
- **Data Localization:** Be aware of data localization requirements, which may mandate that data be stored in specific geographic locations.

7. Usability

- **Intuitive Schema Design:** Design database schemas to be intuitive and easy to understand for developers and users. Use meaningful table and field names.
- **User Interface Integration:** Ensure the database design supports seamless integration with the CRM's user interface, providing a smooth user experience.

8. Flexibility and Extensibility

- **Modular Design:** Design the database schema to be modular, allowing for easy addition of new features and functionality without major restructuring.
- **Custom Fields:** Allow for custom fields and entities to support different business needs and workflows.

11. Disaster Recovery

- **Failover Mechanisms:** Implement failover mechanisms to switch to a backup system in case of a primary system failure.
- **Recovery Point Objectives (RPO) and Recovery Time Objectives (RTO):** Define and implement RPO and RTO to ensure data recovery meets business requirements.

5.Unit Testing

Unit testing is a critical practice in software development, including for cloud-based Customer Relationship Management (CRM) systems. It involves testing individual components or units of code to ensure they function correctly. Here are key considerations and practices for unit testing in a cloud-based CRM:

Key Considerations

1. **Isolation**: Each unit test should be independent of other tests. Mocking and stubbing are often used to isolate the unit being tested from external dependencies like databases or network services.
2. **Automated Testing**: Automation is essential to run tests frequently and consistently. Automated unit tests can be integrated into continuous integration (CI) pipelines to ensure new changes do not break existing functionality.
3. **Test Coverage**: Aim for high test coverage to ensure most parts of the code are tested. However, focus on critical components first to ensure they are robust.
4. **Performance**: Unit tests should be fast. They should test small pieces of code and run quickly to provide immediate feedback.

5. **Maintainability**: Write clear, understandable, and maintainable test cases. This makes it easier to update tests as the code evolves.

Practices

1. **Use of Testing Frameworks**
   - **JUnit (Java)**: A popular testing framework for Java applications.
   - **pytest (Python)**: A framework that makes it easy to write simple and scalable test cases.
   - **JUnit for Spring Boot**: Spring Boot applications often use JUnit along with Spring's testing support.
2. **Mocking Frameworks**
   - **Mockito (Java)**: Used for creating mock objects.
   - **unittest.mock (Python)**: A library for testing in Python.
   - **Sinon.js (JavaScript)**: For mocking and spying in JavaScript.
3. **Test-Driven Development (TDD)**
   - Write tests before writing the actual code.
   - This approach ensures that the code meets the requirements from the start.
4. **Continuous Integration (CI)**
   - Integrate unit tests into CI pipelines using tools like Jenkins, GitHub Actions, or GitLab CI.
   - Automated tests run with every code change to catch issues early.

6.Keyword

Revolutionize your business with our cloud-based CRM. Enhance customer interactions, streamline operations, and make data-driven decisions. Experience seamless integration, improved efficiency, and unparalleled customer satisfaction. Elevate your business today

7.Refrences

1. "CRM at the Speed of Light: Social CRM Strategies, Tools, and Techniques for Engaging Your Customers" by Paul Greenberg

2. "Building a Digital Future: How Cloud Platforms Are Transforming Companies" by Abhinav Garg and Manish Prakash

3. "The CRM Handbook: A Business Guide to Customer Relationship Management" by Jill Dyché.

4. "Cloud-Based CRM Systems: A Review" by International Journal of Scientific and Research Publications.

5. "The Impact of Cloud Computing on Customer Relationship Management Systems" by Journal of Information Systems and Technology Management

6. "Evaluating the Benefits of Cloud-Based CRM Systems for Small and Medium Enterprises" by IEEE Xplore.