Project Design Phase

Technology Stack

Date	01 NOV 2025
Team ID	NM2025TMID04631
Title	CRM Application For Jewel Customer
Maximum Marks	4 Marks

1. Overview

The **CRM Application for Jewel Customer** will be developed using a modern and reliable technology stack to ensure **performance, security, scalability, and ease of maintenance**. The stack combines both **frontend and backend technologies**, as well as **database, integration, and deployment tools**.

---

2. Technology Stack Summary

Layer	Technology / Tool	Purpose
Frontend (User Interface)	HTML5, CSS3, JavaScript	Structure, style, and interactivity for the CRM web interface.
	React.js / Salesforce Lightning Components	For creating responsive, dynamic, and reusable UI components.
Backend (Server-Side)	Node.js with Express / Apex (if on Salesforce platform)	Server logic, API handling, and business process automation.
Database	MySQL / PostgreSQL / Salesforce Object Database	To store customer, order, billing, and inventory data securely.
API Layer	RESTful APIs	To integrate modules like billing, payment, and SMS/email gateways.
Authentication & Security	JWT / OAuth 2.0 / Salesforce Authentication	User login, session management, and secure access control.
Hosting / Deployment	Salesforce Cloud / AWS / Heroku	For reliable cloud-based hosting and deployment.
Version Control	Git, GitHub	For source code management and team collaboration.

Layer	Technology / Tool	Purpose
Communication Services	Twilio / SendGrid / Salesforce Email Service	For sending SMS, email notifications, and alerts.
Payment Integration	Razorpay / Stripe / PayPal APIs	To manage online payments and transaction tracking.
Testing Frameworks	Jest / Mocha / Salesforce Test Classes	Unit and integration testing to ensure system reliability.
Analytics & Reporting	Power BI / Tableau / Salesforce Reports	For generating dashboards, visual analytics, and performance reports.
Containerization (Optional)	Docker	To containerize and deploy the CRM for consistent environments.

---

3. Technology Stack Architecture

The system follows a **3-Tier Architecture**:

1. **Presentation Layer (Frontend/UI)**
    - Developed using **React.js or Salesforce Lightning**.
    - Provides intuitive interfaces for admins, sales teams, and customers.
2. **Application Layer (Backend/Logic)**
    - Implements all **business logic**, workflow automation, and validations.
    - Uses **Node.js/Express** or **Salesforce Apex** for processing requests and managing APIs.
3. **Data Layer (Database/Storage)**
    - Uses **MySQL/PostgreSQL** or **Salesforce Database Objects** to store and retrieve all system data.
    - Includes **data backup and encryption** for reliability and security.

---

4. Integration Components

- **SMS & Email Services:** Twilio / SendGrid APIs for communication.
- **Payment Gateways:** Integration with Razorpay or Stripe for transactions.

- **External APIs:** Support for future integrations like accounting or loyalty programs.
- **Data Analytics:** Integration with Power BI or Salesforce Dashboard for insights.

---

5. Development Tools

Category	**Tools / Software**
IDE / Code Editor	Visual Studio Code / Salesforce Developer Console
Version Control	GitHub / GitLab
Project Management	Jira / Trello
Testing & QA	Postman (API testing), Jest, Selenium
CI/CD	GitHub Actions / Jenkins
Documentation	Confluence / Notion / Salesforce Documentation Tools

---

6. Security Technologies

- SSL/TLS Encryption for all data transmission.
- Role-Based Access Control (Admin, Manager, Sales Executive).
- Data encryption at rest using AES standards.
- Regular security audits and backup mechanisms.

---

7. Deployment & Maintenance

- Hosted on **Salesforce Cloud** or **AWS Cloud Infrastructure**.
- Continuous Integration and Continuous Deployment (CI/CD) pipelines for regular updates.
- Automated monitoring using tools like **CloudWatch** or **Salesforce Monitoring Tools**.