1. Prompt for Dashboard Rendering Module Development for Sales CRM System

Create a dynamic, role-based dashboard rendering module for a Sales CRM system, displaying key performance indicators (KPIs) and visual charts based on user roles (Sales Executive, Sales Head) after successful login. Fetch user role and data from the database, render KPIs (e.g., leads, opportunities count) and charts (e.g., sales pipeline, team performance), and handle errors with clear messages. Ensure validations for role and data, follow industry-standard practices (e.g., secure data fetching, responsive design), achieve load times under 2 seconds, and provide a clean, user-friendly, responsive interface for web and mobile, inspired by professional Sales CRM dashboards (e.g., Sales Head Dashboard, Sales Pipeline, Sales Team Performance, Customer Insights).

Requirements

- **Dashboard Rendering Process**:
    - **Access Dashboard**: After login, users are redirected to /dashboard. Fetch user role (Sales Executive, Sales Head) and data from the database using a secure API call with JWT authentication.
    - Customizable Layouts: Allow drag-and-drop widgets (e.g., KPI cards, charts), save layouts (INSERT INTO dashboard_layouts (user_id, layout_json)).
    - Real-Time Collaboration: Share dashboard views via WebSockets, restricting access by role.

    - **Role-Based Rendering**:
        - **Sales Executive**: Display KPIs (e.g., number of leads, opportunities count) and charts (personal sales pipeline, individual performance). Data fetched from database (e.g., SELECT leads, opportunities FROM sales_data WHERE user_id = ?).
        - **Sales Head**: Display KPIs (e.g., team leads, opportunities, pending approvals) and charts (team sales pipeline, team performance, customer insights). Data fetched from database (e.g., SELECT team_leads, team_opportunities, approvals FROM team_data WHERE team_id = ?).
        - **Interface Components**:
            - KPI cards: <div class="kpi-card"> with title (<h3>Leads</h3>) and value (<p>123</p>), styled with white background, gray border.
            - Charts: <canvas id="pipeline-chart"> for pipeline (bar chart) and <canvas id="performance-chart"> for performance (line chart), using a charting library.

        - Error message: Red text above dashboard (<p class="error">...</p>).
        - Export Button: <button>Export Report</button> (PDF/CSV).
        - Drill-Down Modal: <div class="drill-down-modal"> for detailed data.

    - **Fields**:
        - User Role: Required, must be "Sales Executive" or "Sales Head" (predefined in database).
        - KPI Data: Required, numeric values (e.g., leads, opportunities) from database.
    - **Validation**:
        - Server-side: Verify role in database (SELECT role FROM users WHERE user_id = ?). If invalid, return error: "Unauthorized role." Ensure numeric KPI data; if missing/invalid, show error: "Failed to load dashboard data. Try again."
        - Layout Validation: Ensure valid JSON, show: "Invalid dashboard layout." [Error Code: DASH_004]
- **Data Fetching**:
    - Use secure API endpoint (e.g., /api/dashboard/:user_id) with JWT in Authorization header.
    - Fetch role and data in a single query to minimize latency. Cache results in a key-value store (e.g., 5-minute TTL) for performance.
    - On fetch failure (e.g., database timeout), show error: "Failed to load dashboard data. Please try again later."
    - Drill down for detailed reports
- **Chart Rendering**:
    - Sales Pipeline: Bar chart showing stages (e.g., prospect, negotiation, closed) with numeric values.
    - Performance: Line chart showing sales trends over time (e.g., monthly revenue).
    - Customer Insights (Sales Head only): Pie chart showing customer segments (e.g., new, returning).
    - Ensure charts are responsive, with tooltips for data points and legends.
    - Exportable Reports: Generate PDF/CSV reports with KPI/chart data

- **Interface and User Experience**:
  - **Layout**: Grid-based dashboard with KPI cards at the top and charts below. Card width: 300px on desktop, 90% width on mobile. Page background: light gray; cards/charts: white with gray border. Padding: 24px for cards, 16px for charts.
  - **Colors**: KPI titles: gray; values: black; charts: blue for pipeline, green for performance, multi-colored for insights; error messages: red; background: light gray; cards: white.
  - **Text**: Use Inter or Roboto font. Headings: 20px, bold; KPI values: 18px; chart labels: 14px (headings: 18px on mobile).
  - **Elements**: KPI cards in a responsive grid, charts in full-width containers. Error messages above dashboard. Ensure charts resize on mobile (below 640px).
  - **Responsiveness**: Stack cards and charts vertically on mobile. Follow accessibility standards (ARIA labels like aria-label="Sales Pipeline Chart", keyboard navigation, high-contrast text).
  - **Branding**: Add Sales CRM logo at top-left, keep design clean and professional, matching Sales CRM dashboard aesthetics.
- **Error Messages and Validation**:
  - **Role Validation**: If role is missing/invalid, show: "Invalid user role. Contact support."
  - **Data Validation**: If KPI data is non-numeric or missing, show: "Failed to load dashboard data. Try again."
  - **Fetch Failure**: On database or API failure, show: "Failed to load dashboard data. Please try again later."
  - **Server-Side**: Reject invalid role or data queries, return: "Unauthorized role" or "Invalid data format."
- **Industry Standards**:
  - **Security**: Use HTTPS, validate JWT for API calls, sanitize database inputs to prevent SQL injection, log access without sensitive data.
  - **Performance**: Cache data in a key-value store, use efficient queries, ensure charts render smoothly.
  - **Data Handling**: Fetch only required data, use parameterized queries for security.
- **Non-Functional Requirements**:
  - **Performance**: Load dashboard in under 2 seconds. Cache data, optimize chart rendering.

  - **Scalability**: Support 10,000 users with efficient data fetching and caching.
  - **Reliability**: Ensure 99.9% uptime, retry failed queries 3 times (1s, 2s, 4s delays).
  - **Real-Time Updates**: Poll API every 30 seconds for KPI/chart updates.
  - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
  - **Monitoring**: Log data fetch errors and dashboard access, track load times and error rates.
- **Scenarios**:
  - Sales Executive: Sees personal leads, opportunities, pipeline, and performance charts.
  - Sales Head: Sees team KPIs, approvals, team pipeline, performance, and customer insights.
  - Data Failure: Shows error message and retries fetch.
  - Edge Cases: Invalid role (error displayed), missing data (error shown), slow API (timeout error).

Sub-Tasks

- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

2. Prompt for Company Registration and Validation Module Development for Swayatta 4.0 CRM/ERP System

Develop a robust company registration and validation module for the Swayatta 4.0 CRM/ERP system, enabling Sales Executives and Admins to create and manage company records with validation, audit trails, and role-based access control (RBAC), per the Business Requirement Document (BRD). Replace the removed approval flow with a validation process to classify companies as "cold" (low business potential) or "hot" (high business potential) using a scoring algorithm. Implement cascading dropdowns for country, state, and city, ensure immediate updates to the lead section's company dropdown, and resolve a React state management issue causing city dropdown rendering failures. Include form components, checklists, error messages, and validations. Follow industry-standard CRM practices (e.g., secure data handling, responsive design), ensure encryption, achieve load times under 2 seconds, and provide a clean, user-friendly, responsive interface inspired by Company Management wireframes.

Requirements

- **Company Registration Process**:
  - **Access Form**: Sales Executives/Admins (verified via RBAC) access /company/add to create a company. Restrict access to admin role only, checked server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Admin role required.”
  - Approval Workflow for Updates: Require approval for company updates (/company/edit/:id), notifying Admins (<button>Approve Update</button>).
  - Multi-Language Support: Support Unicode for company name/address, localized dropdowns.
  - **Interface Components**:
    - Form: <form> with inputs for:
      - Company Name: <input type="text" aria-label="Company Name">.
      - GST Number: <input type="text" aria-label="GST Number"> (Indian companies).
      - PAN Number: <input type="text" aria-label="PAN Number">.
      - Company Type: <select aria-label="Company Type"> (e.g., Private, Public).
      - Account Type: <select aria-label="Account Type"> (e.g., Customer, Partner).
      - Region: <select aria-label="Region"> (e.g., APAC, EMEA).
      - Business Type: <select aria-label="Business Type"> (e.g., B2B, B2C).
      - Industry: <select aria-label="Industry"> (e.g., IT_ITeS, BFSI, Healthcare, Manufacturing, Energy_Utilities, Telecom).
      - Sub-Industry: <select aria-label="Sub-Industry"> (e.g., Software, Banking).
      - Billing Record ID: <input type="number" aria-label="Billing Record ID">.
      - Website: <input type="text" aria-label="Website"> (optional).
      - Is Child: <input type="checkbox" aria-label="Is Child Company">.
      - Parent Name: <select aria-label="Parent Company"> (conditional, shown if Is Child checked).
      - Address: <textarea aria-label="Address">.

        - Employee Count: <input type="number" aria-label="Employee Count">.
        - Country/State/City: Cascading <select> dropdowns (aria-label="Country", State, City).
        - Submit: <button>Save Company</button>.
        - Annual Revenue: <input type="number" aria-label="Annual Revenue">, <select aria-label="Currency"> (e.g., INR, USD, EUR).
        - Bulk Import: <input type="file" aria-label="Import CSV">.
    - Checklist: <div class="checklist"> with Go/No-Go items (e.g., <input type="checkbox" aria-label="Valid GST"> for GST verified, <input type="checkbox" aria-label="Active Status"> for active).
    - Error message: Red text above form (<p class="error">...</p>).
    - Success message: Green text above form (<p class="success">Company added successfully</p>).
- **Fields**:
    - Company ID: Auto-generated, unique integer.
    - Company Name: Required, alphanumeric, 3-100 characters, unique.
    - GST Number: Required for Indian companies (15-digit alphanumeric, e.g., 22AAAAA0000A1Z5), optional otherwise.
    - PAN Number: Required if no GST (10-char alphanumeric, e.g., AAAAA9999A), optional otherwise.
    - Company Type/Account Type/Region/Business Type/Industry/Sub-Industry: Required, from predefined dropdown lists.
    - Billing Record ID: Required, unique numeric.
    - Website: Optional, valid URL format.
    - Is Child: Required, boolean.
    - Parent Name: Required if Is Child is true, dropdown of existing companies.
    - Address: Required, up to 500 characters.
    - Employee Count: Required, integer, default 0.
    - Country/State/City: Required, foreign keys to respective tables.
    - Annual Revenue: Optional, positive numeric, currency (INR, USD, EUR from currencies table)
    - Status: Required, boolean, default active.

    - Created By/On: Required, auto-filled with admin ID and timestamp.
    - Modified By/On: Optional, auto-filled on updates.
    - Remarks: Optional, text.
    - Lead Status: Required, “cold” or “hot” based on validation.
  - **Validation**:
    - Server-side:
      - Duplicate check: Query database (SELECT company_name, gst_number, pan_number FROM companies), show error: “Company name/GST/PAN already exists.”
      - GST/PAN: For India (country_id = India), ensure GST or PAN is provided, show error: “GST or PAN required for Indian companies.”
      - Billing Record ID: Check uniqueness, show error: “Billing Record ID already in use.”
      - Parent Name: If Is Child, verify parent exists, show error: “Invalid parent company.”
      - Currency Validation: Verify currency exists (SELECT id FROM currencies WHERE code = ?), show: “Invalid currency.”

- Bulk Import: Validate CSV format, show: “Invalid CSV format.” Third-Party Data Enrichment:
  - Integrate APIs (e.g., Dun & Bradstreet) to auto-fill revenue, employee count. Show: “Enriched data unavailable.”
- Bulk Import/Export:
  - Import CSV (/company/import), validate duplicates. Export to CSV (/company/export). Show: “Import/Export completed successfully.”
    - **Cascading Dropdowns**:
      - Populate Country dropdown from countries table (SELECT id, name FROM countries).
      - On country selection, populate State dropdown (SELECT id, name FROM states WHERE country_id = ?).
      - On state selection, populate City dropdown (SELECT id, name FROM cities WHERE state_id = ?).
      - Predefined data: Countries (United States, Canada, India); States (e.g., India: Maharashtra, Karnataka, Delhi; United States: California, New York; Canada: Ontario, Quebec); Cities (e.g., Maharashtra: Mumbai, Pune, Nagpur, 19 cities total).
      - Fix React issue: Prevent city dropdown clearing (e.g., setCities([]) after loading). Optimize useEffect dependencies (trigger only on

        state_id), use useCallback for fetch functions, and memoize state updates to avoid re-renders. Show error if no cities load: “Failed to load cities. Try again.”

  - **Validation Process**:
    - Classify as “cold” (<70 score) or “hot” (≥70 score) using algorithm:
      - Industry: 40 points (target industries: IT_ITeS, BFSI, Healthcare, Manufacturing, Energy_Utilities, Telecom; else 10).
      - Sub-Industry: 20 points (relevant sub-industries, else 5).
      - Revenue: 25 points (> $1M or default, else 10).
      - Employee Count: 15 points (> 50, else 5).
    - Store result in lead_status field. Update lead section’s company dropdown immediately (SELECT company_id, company_name FROM companies WHERE status = true).
  - **Checklist and Notification**:
    - Go/No-Go checklist: Verify GST/PAN, active status, and parent linkage (if applicable). Display as checkboxes, require all checked to submit.
    - On save, notify admin via email (branded, with company details). Retry delivery 3 times (1s, 2s, 4s delays). Show error: “Failed to send notification. Company saved.”
  - **Audit Trails**:
    - Log CRUD operations in audit_logs table (id, entity_type=company, entity_id, action, user_id, timestamp, details JSON).
    - Include created_by, modified_by, created_at, modified_at in company record.
    - Versioning: Store company versions
- **Interface and User Experience**:
  - **Layout**: Center a card (400px wide on desktop, 90% on mobile). Page background: light gray; card: white. Padding: 32px for card, 16px for fields.
  - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
  - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
  - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Checklist below form. Error/success messages above form. Submit button at bottom.

    - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Add Swayatta CRM logo above form, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Duplicate Name/GST/PAN: “Company name/GST/PAN already exists.”
        - Missing GST/PAN (India): “GST or PAN required for Indian companies.”
        - Invalid Billing ID: “Billing Record ID already in use.”
        - Invalid Parent: “Invalid parent company.”
        - Database Failure: “Failed to save company. Try again.”
        - Unauthorized Access: “Access denied. Admin role required.”
    - **Checklist**: If unchecked, show: “Complete all checklist items to proceed.”
    - **Notification Failure**: “Failed to send notification. Company saved.”
- **Industry Standards**:
    - **Security**: Use HTTPS, parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
    - **Data Integrity**: Enforce uniqueness constraints, restrict deletion of companies linked to active leads/contacts/opportunities (set status = false).
    - **AI Integration**: Use AI for deduplication by comparing company name, GST, and PAN (e.g., fuzzy matching), flag duplicates for review.
    - Multi-Language: Unicode support, localized dropdowns
- **Non-Functional Requirements**:
    - **Performance**: Load form and save data in under 2 seconds. Cache dropdown data, optimize queries.
    - **Scalability**: Handle 10,000 users with efficient database indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry notifications 3 times, use circuit breakers for external services.
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log CRUD operations and errors, track form submission success rates and validation errors.

- **Scenarios**:
    - New Company: Admin creates company, passes validation, classified as “hot,” added to lead dropdown.
    - Child Company: Admin links to parent, completes checklist, saves successfully.
    - Duplicate Error: Duplicate name/GST flagged, error shown.
    - Invalid Entry: Wrong GST/PAN format, error displayed.
    - City Dropdown Failure: Fixed to show cities correctly.
- **Technical Instructions**:

  Sub-Tasks

- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

3. **Prompt for Contact Management Module Development for Swayatta 4.0 CRM/ERP System**

Project Overview

Develop a secure, efficient contact management module for the Swayatta 4.0 CRM/ERP system, enabling Sales Executives to create and update contact records linked to companies, designate a Single Point of Contact (SPOC), encrypt sensitive data, and sync with the database. Include form components, validations, and error messages for contact creation and updates. Restrict access to Sales Executives via role-based access control (RBAC), implement AI-based duplicate detection, and ensure data integrity. Follow industry-standard CRM practices (e.g., secure data handling, responsive design), achieve load times under 2 seconds, and provide a clean, user-friendly, responsive interface inspired by Contact Management wireframes.

Requirements

- **Contact Creation and Update Process**:
    - **Access Form**: Sales Executives (verified via RBAC) access /contacts/add or /contacts/edit/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Sales Executive role required.”
    - **Interface Components**:
        - Form: <form> with inputs for:
            - Company: <select aria-label="Company"> (populated from companies table, active only).

        - Salutation: <select aria-label="Salutation"> (options: Mr., Ms.).
        - First Name: <input type="text" aria-label="First Name">.
        - Middle Name: <input type="text" aria-label="Middle Name"> (optional).
        - Last Name: <input type="text" aria-label="Last Name"> (optional).
        - Email: <input type="email" aria-label="Email">.
        - Primary Phone: <input type="tel" aria-label="Primary Phone">.
        - Designation: <select aria-label="Designation"> (e.g., Manager, Director, CEO).
        - LinkedIn Profile
        - Decision Maker: <input type="checkbox" aria-label="Decision Maker">.
        - SPOC: <input type="checkbox" aria-label="SPOC"> (only one per company).
        - Comments: <textarea aria-label="Comments"> (optional).
        - Option: <select aria-label="Option"> (optional, predefined list, e.g., Preferred Contact Method).
        - Submit: <button>Save Contact</button>.
        - Activity Timeline: <div class="activity-timeline"> for interactions.
    - Error message: Red text above form (<p class="error">...</p>).
    - Success message: Green text above form (<p class="success">Contact saved successfully</p>).
- **Fields**:
    - Sr. No.: Auto-generated, unique numeric.
    - Salutation: Required, dropdown (Mr., Ms.).
    - First Name: Required, alphabetic, up to 250 characters.
    - Middle Name/Last Name: Optional, alphabetic, up to 250 characters.
    - Email: Required, valid email format.
    - Primary Phone: Required, 10-15 digits.
    - Designation: Required, from predefined dropdown list.

        - Decision Maker: Required, boolean.
        - SPOC: Required, boolean (only one contact per company can be SPOC).
        - Comments: Optional, up to 500 characters.
        - Option: Optional, from predefined dropdown.
        - Created By/On: Required, auto-filled with Sales Executive ID and timestamp.
        - Modified By/On: Optional, auto-filled on updates.
        - LinkedIn Profile: Optional, valid URL.
    - **Company Linking**:
        - Populate Company dropdown from companies table (SELECT company_id, company_name FROM companies WHERE status = true).
        - Link contact to selected company via company_id foreign key.
        - On SPOC designation, check existing SPOC (SELECT id FROM contacts WHERE company_id = ? AND spoc = true). If exists, show error: "Another contact is already SPOC for this company."
    - **Validation**:
        - Server-side:
            - Duplicate check: Use AI (fuzzy matching on email, first name, company_id) to flag duplicates, show error: "Possible duplicate contact detected. Review and confirm."
            - Email: Query database (SELECT id FROM contacts WHERE email = ?), show error: "Email already in use."
            - SPOC: Ensure only one SPOC per company, show error: "Another contact is already SPOC."
            - Database failure: Show error: "Failed to save contact. Try again."
            - LinkedIn Profile: Validate URL, show: "Invalid LinkedIn URL."
- Contact Activity Tracking:
    - Log interactions (INSERT INTO contact_activities (contact_id, type, timestamp)). Display timeline (SELECT * FROM contact_activities WHERE contact_id = ?).
- Bulk Contact Management:
    - Import/export CSV (/contacts/import, /contacts/export). Validate duplicates, show: "Duplicate email detected in import."
        - **Data Encryption**:

        - Encrypt sensitive fields (Email, Primary Phone) using AES-256 before saving (INSERT INTO contacts).
        - Store encryption keys securely in environment variables.
        - Decrypt for display/edit using secure API endpoint with JWT authentication.
    - **Save and Sync**:
        - Save contact to contacts table with audit fields (created_by, created_at, modified_by, modified_at).
        - Sync updates to lead section via real-time API (/api/contacts/sync/:company_id).
        - On save, show success: "Contact saved successfully." On failure, show error: "Failed to save contact. Try again."
- **Interface and User Experience**:
    - **Layout**: Center a card (400px wide on desktop, 90% on mobile). Page background: light gray; card: white. Padding: 32px for card, 16px for fields.
    - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
    - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
    - **Form Elements**: Full-width inputs/dropdowns, rounded corners. SPOC checkbox prominent. Error/success messages above form. Submit button at bottom.
    - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Add Swayatta CRM logo above form, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Duplicate Contact: "Possible duplicate contact detected. Review and confirm."
        - Email in Use: "Email already in use."
        - SPOC Conflict: "Another contact is already SPOC for this company."
        - Database Failure: "Failed to save contact. Try again."
        - Unauthorized Access: "Access denied. Sales Executive role required."
    - **Sync Failure**: "Failed to sync contact with lead section. Try again."
- **Industry Standards**:

  - **Security**: Use HTTPS, AES-256 encryption for sensitive data, parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
  - **Data Integrity**: Enforce SPOC uniqueness per company, prevent deletion of contacts linked to active leads/opportunities.
  - **AI Integration**: Use fuzzy matching for duplicate detection (email, first name, company_id), flag for review.
- **Non-Functional Requirements**:
  - **Performance**: Load form and save data in under 2 seconds. Cache company dropdown data, optimize queries.
  - **Scalability**: Handle 10,000 users with efficient indexing and caching.
  - **Reliability**: Ensure 99.9% uptime, retry sync operations 3 times (1s, 2s, 4s delays).
  - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
  - **Monitoring**: Log CRUD operations and errors, track form submission success rates and duplicate flags.
- **Scenarios**:
  - New Contact: Sales Executive creates contact, links to company, designates SPOC, saves successfully.
  - Update Contact: Updates existing contact, syncs with lead section.
  - Duplicate Contact: AI flags duplicate email, error shown.
  - SPOC Conflict: Attempt to set multiple SPOCs fails, error displayed.
  - Edge Cases: Invalid email/phone (error shown), sync failure (retry, then error).
- **Technical Instructions**:

  Sub-Tasks
- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

4. Prompt for Lead Management Module Development for Swayatta 4.0 CRM/ERP System

Project Overview

Develop a comprehensive lead management module for the Swayatta 4.0 CRM/ERP system, enabling Sales Executives to capture, nurture, and convert leads into opportunities, with support for domestic and international leads. Include tender type selection (Tender, Pre-Tender, Post-Tender, Non-Tender), company/partner validation, proof uploads, Go/No-Go checklists, and notification integration. Restrict access to Sales Executives via role-based access control (RBAC), validate lead details, and escalate conflicts. Follow industry-standard CRM practices (e.g., secure data handling, responsive design), achieve load times under 2 seconds, and provide a clean, user-friendly, responsive interface inspired by Lead List wireframes. The module must adapt form fields based on tender type, disabling irrelevant fields for Non-Tender leads as per the attached photo concept.

Requirements

- **Lead Capture, Nurture, and Conversion Process**:
  - **Access Form**: Sales Executives (verified via RBAC) access /leads/add or /leads/edit/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Sales Executive role required.”
  - **Interface Components**:
    - Form: <form> with inputs for:
      - Tender Type: <select aria-label="Tender Type"> (options: Tender, Pre-Tender, Post-Tender, Non-Tender).
      - Lead ID: Auto-generated alphanumeric (display only).
      - Project Title: <input type="text" aria-label="Project Title">.
      - Company Name: <select aria-label="Company Name"> (populated from companies table, active only).
      - State: <select aria-label="State"> (predefined list, e.g., Maharashtra, Karnataka).
      - Partner: <select aria-label="Partner"> (optional, from partners table).
      - Lead Subtype: <select aria-label="Lead Subtype"> (e.g., Direct, Referral).
      - Source: <select aria-label="Source"> (e.g., Website, Event).
      - Product: <select aria-label="Product"> (from catalog table).
      - Is Enquiry: <input type="checkbox" aria-label="Is Enquiry">.
      - Billing Type: <select aria-label="Billing Type"> (e.g., Fixed, Variable, international only).
      - Expected ORC: <input type="number" aria-label="Expected ORC"> (international only).

      - Status: <select aria-label="Status"> (e.g., New, Nurturing, Converted).
      - Lead/Account Owner: <select aria-label="Lead Owner"> (from users table).
      - Approval Status: <select aria-label="Approval Status"> (e.g., Pending, Approved, Rejected).
      - Upload Proofs: <input type="file" aria-label="Upload Proofs" multiple> (e.g., PDFs, images).
      - Go/No-Go Checklist: <div class="checklist"> with items (e.g., <input type="checkbox" aria-label="Validated Company">).
      - Submit: <button>Save Lead</button>.
    - Error message: Red text above form (<p class="error">...</p>).
    - Success message: Green text above form (<p class="success">Lead saved successfully</p>).
  - **Fields**:
    - Lead ID: Auto-generated alphanumeric.
    - Project Title: Required, up to 255 characters, no special characters.
    - Company Name: Required, from existing companies.
    - State: Required, from predefined list.
    - Partner: Optional, from predefined list.
    - Status: Required, from predefined list (e.g., New, Nurturing, Converted).
    - Created On/By: Auto-filled with timestamp and Sales Executive ID.
    - Lead/Account Owner: Required, from user list.
    - Approval Status: Required, from predefined list (e.g., Pending, Approved, Rejected).
    - Lead Subtype/Source: Required, from dropdowns.
    - Product: Required, from catalog.
    - Is Enquiry: Required, boolean.
    - Billing Type: Required for international, from dropdown (e.g., Fixed, Variable).
    - Expected ORC: Required for international, numeric.
  - **Tender Type Logic**:
    - If Tender/Pre-Tender/Post-Tender selected (as per attached photo):

        - Enable Billing Type and Expected ORC fields.
        - Require completion for submission.
      - If Non-Tender selected:
        - Disable and hide Billing Type and Expected ORC fields.
        - Mark as optional, allow submission without these fields.
      - Show error if mismatch: “Billing Type and ORC required for Tender/Pre-Tender/Post-Tender.”
    - **Validation**:
      - Server-side:
        - Company/Partner validation: Check existence (SELECT id FROM companies/partners WHERE id = ?), show error: “Invalid company or partner.”
        - Conflict check: Compare lead details (company, project title) with existing leads, escalate if conflict, show error: “Lead conflict detected. Escalated for review.”
        - File upload: Validate file type (PDF, image), size (<5MB), show error: “Invalid file type or size.”
        - Database failure: Show error: “Failed to save lead. Try again.”
        - Currency: Verify exists (SELECT id FROM currencies WHERE code = ?), show: “Invalid currency.”

- Automated Lead Scoring:
  - Score leads based on behavior (e.g., email opens, website visits, SELECT engagement_score FROM lead_activities WHERE lead_id = ?). Update status dynamically.
- Lead Assignment Rules:
  - Configure rules (e.g., round-robin, region-based) via /leads/rules. Show: “Invalid assignment rule.”
- Nurturing Workflows:
  - Automate drip email campaigns via integration (e.g., SendGrid). Track engagement, show: “Failed to trigger nurturing campaign.”

      -

    - **Go/No-Go Checklist**:
      - Items: Validate company, verify proofs, confirm status. Display as checkboxes, require all checked to submit.
      - Show error if unchecked: “Complete all checklist items to proceed.”
    - **Nurture and Conversion**:

        - Nurture: Update status to "Nurturing" with notes, notify owner via email.
        - Conversion: On approval (Approval Status = Approved), convert to opportunity (INSERT INTO opportunities), notify team. On rejection, escalate with error: "Lead rejected. Escalated for review."
    - **Notifications**:
        - Send branded email on save/nurture/conversion (e.g., "Lead updated, action required"). Retry 3 times (1s, 2s, 4s delays). Show error: "Failed to send notification. Lead saved."
- **Interface and User Experience**:
    - **Layout**: Center a card (400px wide on desktop, 90% on mobile). Page background: light gray; card: white. Padding: 32px for card, 16px for fields.
    - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
    - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
    - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Checklist below form. Tender-specific fields toggle visibility. Error/success messages above form. Submit button at bottom.
    - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Add Swayatta CRM logo above form, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Invalid Company/Partner: "Invalid company or partner."
        - Lead Conflict: "Lead conflict detected. Escalated for review."
        - Database Failure: "Failed to save lead. Try again."
        - Notification Failure: "Failed to send notification. Lead saved."
        - Unauthorized Access: "Access denied. Sales Executive role required."
    - **Checklist**: "Complete all checklist items to proceed."
- **Industry Standards**:
    - **Security**: Use HTTPS, encrypt sensitive data (e.g., email), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).

    - **Data Integrity**: Enforce company/partner existence, prevent deletion of converted leads.
    - **Notifications**: Use secure email transport (TLS), retry logic for reliability.
- **Non-Functional Requirements**:
    - **Performance**: Load form and save data in under 2 seconds. Cache dropdown data, optimize queries.
    - **Scalability**: Handle 10,000 users with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry notifications 3 times, use circuit breakers for external services.
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log CRUD operations and errors, track lead conversion rates and conflict escalations.
- **Scenarios**:
    - Domestic Lead: Sales Executive creates lead, validates company, converts to opportunity.
    - International Lead: Includes Billing Type/ORC, nurtured, approved.
    - Partner Involved: Partner selected, checklist completed, saved.
    - Conflict: Duplicate lead detected, escalated.
    - Edge Cases: Invalid data (error shown), notification failure (retry, then error).
- **Technical Instructions**:

  Sub-Tasks
- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

5. Prompt for Opportunity Management Module Development for Swayatta 4.0 CRM/ERP System

Project Overview

Develop a robust opportunity management module for the Swayatta 4.0 CRM/ERP system, enabling Sales Executives and Managers to manage the sales pipeline through stages (L1 to L8) by directly entering all details server-side, converting leads or creating opportunities manually, updating stages, uploading documents/competitors/clauses, requesting approvals, and marking opportunities as won or lost. Include stage-specific fields, mandatory validations, approval workflows, internal-only fields (e.g., CPC, Overhead), and auto-transition logic. Enhance the L4 Technical Qualification (Quotation) stage with integrated proposal creation and versioning under the Opportunity ID, allowing Sales Executives and Solutions Teams to enter details, add groups/phases, version changes, secure with digital signatures, and approve/send. Restrict access via role-based access control (RBAC), enforce stage progression rules, and integrate notifications for escalations/delays. Follow industry-standard CRM practices (e.g., secure data handling, responsive design), achieve load times under 2 seconds, and provide a clean, user-friendly, responsive interface inspired by Opportunity List and Quotation Details wireframes.

Requirements

- **Opportunity Pipeline Management Process**:
    - **Access Form**: Sales Executives/Managers (verified via RBAC) access /opportunities/add or /opportunities/edit/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Sales Executive/Manager role required.”
    - **Interface Components**:
        - Form: <form> with stage-specific inputs (dynamic based on stage, filled by Salesperson):
            - Opportunity ID: Auto-generated alphanumeric (display only, used for proposal/quotation).
            - Status: <select aria-label="Status"> (e.g., Proposal, Won, Lost, stages L1-L8).
            - Document: <input type="file" aria-label="Upload Document" accept="application/pdf,image/jpeg" multiple> (max 5MB).
            - Competitor Name: <input type="text" aria-label="Competitor Name"> (optional, 100 chars).
            - Expected Revenue: <input type="number" aria-label="Expected Revenue" min="0">.
            - Currency: <select aria-label="Currency"> (e.g., INR, USD).
            - Convert Date: <input type="date" aria-label="Convert Date">.
            - Stage-specific fields (see below).
            - Submit: <button>Save Opportunity</button>.
        - Pipeline View: <div class="pipeline"> with stage cards (e.g., L1 Prospect, L4 Technical Qualification).
        - Error message: Red text above form (<p class="error">...</p>).

        - Success message: Green text above form (<p class="success">Opportunity saved successfully</p>).
        - Forecasting Dashboard: <div class="forecast"> with weighted revenue.
        - Competitor Analysis: <div class="competitor-analysis"> with charts.
    - **Stages and Fields**:
        - **L1 Prospect**:
            - Fields: Manual/Enriched Data (text), Region (dropdown, e.g., APAC, EMEA), Product Interest (dropdown from catalog), Assigned Rep (dropdown from users), Status (dropdown, e.g., New), Notes (text, 500 chars).
            - Mandatory: Region, Product Interest, Assigned Rep.
            - Action: Capture lead details, auto-transition to L2 if data validated.
        - **L2 Qualification**:
            - Fields: Scorecard (BANT/CHAMP: Budget, Authority, Need, Timeline, text), Status (dropdown: Qualified, Not Now, Disqualified), Go/No-Go Checklist (<div class="checklist"> with items).
            - Mandatory: Scorecard, Status.
            - Action: Complete checklist, auto-transition to L3 if Qualified.
            - Validation: Block if checklist incomplete, show error: “Complete Go/No-Go checklist.”
        - **L3 Proposal/Bid**:
            - Fields: Proposal Upload (<input type="file">), Version (text), Submission Date (<input type="date">), Internal Stakeholders (dropdown from users), Client Response (text).
            - Mandatory: Proposal Upload, Submission Date.
            - Action: Request approval, escalate if delayed.
        - **L4 Technical Qualification (Quotation)**:
            - Fields: Proposal Creation (under Opportunity ID, Group/Phase: dropdown, e.g., Hardware, Software, Service), Item Selection (dropdown with qty, rate), Quotation Status (dropdown: Draft, Submitted, Approved), Version (auto-increment), Digital Signature (<input type="file" aria-label="Digital Signature">, valid format), Approval Workflow (<button>Request Approval</button>), Revision Comments (text), Version History (display only).

        - Mandatory: Group/Phase, Item Selection, Quotation Status, Digital Signature.
        - Action: Enter details, add groups/phases, version changes, secure with digital signature, approve/send. Lock until approved, auto-transition to L5 on approval.
        - Validation: Block send if unapproved, show error: “Cannot send unapproved proposal.”
    - **L5 Commercial Negotiations**:
        - Fields: Updated Pricing (numeric), Margin Check (numeric), CPC (internal, hidden), Overhead (internal, hidden), Commercial Approval (<button>Request Approval</button>), T&C (text), Legal Doc Status (dropdown), PO Fields (number, date, amount, file, status).
        - Mandatory: Updated Pricing, Margin Check, PO Number.
        - Action: Request commercial approval, escalate if delayed.
    - **L6 Won**:
        - Fields: Final Value (numeric), Client PoC (text), Handover Status (dropdown), Delivery Team (dropdown from users), Kickoff Task (text), Revenue Recognition Flag (checkbox).
        - Mandatory: Final Value, Handover Status.
        - Action: Assign team, mark as won, notify delivery team.
    - **L7 Lost**:
        - Fields: Lost Reason (dropdown, e.g., Price, Competitor), Competitor (text), Follow-Up Reminder (<input type="date">), Internal Learnings (text), Escalation Flag (checkbox).
        - Mandatory: Lost Reason.
        - Action: Escalate if flagged, set reminder.
    - **L8 Dropped**:
        - Fields: Drop Reason (dropdown, e.g., Inactive), Auto-Drop Logic (trigger if inactive 30 days), Re-Nurture Tag (checkbox), Reminder Date (<input type="date">), Reactivation Option (button).
        - Mandatory: Drop Reason.
        - Action: Auto-drop if inactive, set reminder for reactivation.
- **Validation**:
    - Server-side (all validations performed by Salesperson input):

- Expected Revenue: Check positive numeric (/^\d+(\.\d{1,2})?$/), show error: “Expected revenue must be positive.”
- Convert Date: Check not past (e.g., > 2025-08-27), show error: “Convert date cannot be in the past.”
- Document: Validate type/size (PDF/JPG, <5MB), show error: “Invalid file type or size exceeds 5MB.”
- Digital Signature: Validate format (e.g., PDF with signature), show error: “Invalid digital signature format.”
- Stage skip: Block progression if previous stage incomplete, show error: “Complete previous stage before proceeding.”
- Duplicate check: Flag if opportunity ID conflicts, show error: “Opportunity ID already exists.”
- Mandatory fields: Prevent submission if empty, show error: “All required fields must be filled.”
- Unapproved send: Block proposal send in L4, show error: “Cannot send unapproved proposal.”
- Approval delay: Escalate if >48 hours, show error: “Approval delayed. Escalated.”

- **Approval Workflow**:
  - Request approval via button, notify manager via email. On approval, auto-transition; on rejection, escalate with error: “Approval rejected. Escalated for review.”
- **Notifications**:
  - Send branded email on save/approval/escalation/send (e.g., “Opportunity requires action”). Retry 3 times (1s, 2s, 4s delays). Show error: “Failed to send notification. Opportunity saved.”

- **Interface and User Experience**:
  - **Layout**: Pipeline view with stage columns (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 24px for cards, 16px for fields.
  - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
  - **Text**: Inter or Roboto font. Headings: 20px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
  - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Pipeline cards draggable for stage updates. Internal fields (CPC, Overhead) hidden with CSS. Error/success messages above form. Submit button at bottom.

    - **Responsiveness**: Stack stages vertically on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Add Swayatta CRM logo at top-left, keep design clean and professional.
  - **Error Messages and Validation**:
    - **Server-Side**:
      - Empty fields: “All required fields must be filled.”
      - Invalid Revenue: “Expected revenue must be positive.”
      - Invalid Date: “Convert date cannot be in the past.”
      - Invalid File: “Invalid file type or size exceeds 5MB.”
      - Invalid Signature: “Invalid digital signature format.”
      - Stage Skip: “Complete previous stage before proceeding.”
      - Duplicate ID: “Opportunity ID already exists.”
      - Unapproved Send: “Cannot send unapproved proposal.”
      - Approval Delay: “Approval delayed. Escalated.”
      - Approval Rejection: “Approval rejected. Escalated for review.”
      - Database Failure: “Failed to save opportunity. Try again.”
      - Unauthorized Access: “Access denied. Sales Executive/Manager role required.”
      - Currency: Verify exists (SELECT id FROM currencies WHERE code = ?), show: “Invalid currency.”
- Forecasting Tools:
  - Calculate weighted revenue (SELECT SUM(expected_revenue * probability) FROM opportunities WHERE status = ?). Display in dashboard.
- Competitor Analysis:
  - Aggregate data (SELECT competitor_name, count(*) FROM opportunities GROUP BY competitor_name). Display bar chart.
- Automated Stage Transitions:
  - Trigger transitions on events (e.g., document upload). Show: “Stage transition failed.”

  - **Industry Standards**:
    - **Security**: Use HTTPS, encrypt sensitive data (e.g., revenue), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent

XSS, log actions securely (no sensitive data), secure digital signatures with PKI.

    - **Data Integrity**: Enforce stage progression, prevent deletion of won/lost opportunities.
    - **Notifications**: Use secure email transport (TLS), retry logic for reliability.

- **Non-Functional Requirements**:
    - **Performance**: Load pipeline and save data in under 2 seconds. Cache stage data, optimize queries.
    - **Scalability**: Handle 10,000 users with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry notifications 3 times, use circuit breakers for external services.
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log CRUD operations and errors, track stage transitions and approval delays.
- **Scenarios**:
    - From Lead: Convert lead to L1 Prospect, progress to L6 Won.
    - Stage Progression: Move from L2 to L4, create proposal under Opportunity ID, request approvals.
    - Approvals: L4 approved with signature, L5 escalated on delay.
    - Edge Cases: Invalid date/revenue (error shown), duplicate ID (error flagged), unapproved send (blocked), sync failure (retry, then error).
- **Technical Instructions**:

Sub-Tasks

- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

6. Prompt for Profitability Visualization Module Development for Swayatta 4.0 CRM/ERP System

Develop a profitability visualization module for the Swayatta 4.0 CRM/ERP system, integrated into the L4 Technical Qualification (Quotation) stage of the opportunity management process. This module will automatically generate a profit/loss analysis as soon as a quotation is prepared, using purchase costs stored in the backend database and selling price data from the quotation. Display the analysis in a table below the quotation form, with color-coded profit/loss indicators (red for negative profit), and provide an exportable PnL template in Excel format for the specific opportunity. Ensure the module is accessible only to Sales Executives and Managers via role-based access control (RBAC), achieves load times under 2 seconds, and follows industry-standard practices (e.g., secure data handling, responsive design).

Requirements

- **Profitability Visualization Process**:
  - **Access**: Sales Executives/Managers (verified via RBAC) access profitability data at /opportunities/edit/:id during L4. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Sales Executive/Manager role required.”
  - **Interface Components**:
    - Profitability Table: <table class="pnl-table"> below quotation form in L4, displaying:
      - Sr. No., Product Name, SKU Code, Qty, Unit, Purchase Cost (Cost per Unit, Total), Selling Price (Price List per Unit, Selling Rate per Unit, Discount Percentage, Total), Phase Total, Grand Total, Summary (Total One Time Cost, Total Recurring Purchase Cost, External Purchase Cost, Total Tenure, Total Project Cost, Total Selling Price, Total Project Profit [red if negative], Profit Percentage).
      - Currency: <select aria-label="Currency"> (INR, USD, EUR).
    - Export PnL: <button>Export PnL Template</button> generates Excel file with PnL data for the opportunity.
    - What-If Analysis: <input type="number" aria-label="Hypothetical Discount">, <button>Recalculate Profit</button>.
    - Historical Trends: <canvas id="profit-trend-chart">.
    - Error message: Red text above table (<p class="error">...</p>).
    - Success message: Green text above table (<p class="success">PnL exported successfully</p>).
  - **Data Sources**:
    - Purchase Cost: Stored in backend database (opportunities table, e.g., purchase_cost field).
    - Selling Price: Derived from quotation data (Group/Phase, Item Selection, qty, rate).
    - Calculation: Total Project Profit = Total Selling Price - Total Project Cost; Profit Percentage = (Total Project Profit / Total Selling Price) * 100.

        - Convert costs/prices to selected currency using exchange rates (SELECT rate FROM exchange_rates WHERE currency_code = ?).
    - **Validation**:
        - Server-side (all validations performed by Salesperson input):
            - Missing purchase cost: Show error: “Purchase cost data missing for profitability calculation.”
            - Invalid quotation data: Show error: “Quotation data incomplete for profitability analysis.”
            - Negative profit: Highlight Total Project Profit in red in table.
            - Database failure: Show error: “Failed to generate profitability data. Try again.”
            - Currency: Verify exists (SELECT id FROM currencies WHERE code = ?), show: “Invalid currency.”

- **Interface and User Experience**:
    - **Layout**: Table below quotation form (400px wide on desktop, 90% on mobile). Page background: light gray; table: white. Padding: 16px for table cells.
    - **Colors**: Table headers: gray; profit: green; loss: red; labels: dark gray; inputs: white with gray border.
    - **Text**: Inter or Roboto font. Headers: 18px, bold; data: 14px (headers: 16px on mobile).
    - **Form Elements**: Responsive table with scroll on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Align with Swayatta CRM logo, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Missing Purchase Cost: “Purchase cost data missing for profitability calculation.”
        - Invalid Quotation Data: “Quotation data incomplete for profitability analysis.”
        - Database Failure: “Failed to generate profitability data. Try again.”
        - Unauthorized Access: “Access denied. Sales Executive/Manager role required.”
- **Industry Standards**:

  - **Security**: Use HTTPS, encrypt sensitive data (e.g., purchase costs), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
  - **Data Integrity**: Ensure purchase costs and quotation data are consistent, prevent manual overrides during L4.
  - **Export**: Generate Excel with standardized PnL format, secure file generation with temporary URLs.
- **Non-Functional Requirements**:
  - **Performance**: Load profitability table and export in under 2 seconds. Cache purchase cost data, optimize queries.
  - **Scalability**: Handle 10,000 opportunities with efficient indexing and caching.
  - **Reliability**: Ensure 99.9% uptime, retry export generation 3 times (1s, 2s, 4s delays).
  - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
  - **Monitoring**: Log profitability calculations and export actions, track error rates.
- **Scenarios**:
  - New Quotation: Prepare quotation in L4, visualize profit (e.g., ₹275,379.6) with green indicator, export PnL.
  - Loss Scenario: Quotation with negative profit (e.g., -₹10,000) shows red indicator.
  - Edge Cases: Missing purchase cost (error shown), invalid data (error flagged), export failure (retry, then error).
- **Technical Instructions**:

Sub-Tasks

- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

7. Prompt for CRM Delivery and Billing Integration Module Development for Swayatta 4.0 CRM/ERP System

Develop a delivery and billing integration module for the Swayatta 4.0 CRM/ERP system, extending visibility into delivery and billing milestones from the L6 Won stage onward, aligning with modern CRM practices (e.g., Salesforce, Dynamics, Zoho, HubSpot with CPQ modules). Enable Sales Executives and Managers to track product delivery progress, billing status, and invoice details synced read-only from an ERP system (e.g., SAP, Oracle, Dynamics, Tally), addressing customer expectations, improving forecasting accuracy, and supporting account management/upsell opportunities. Restrict editing to maintain data governance, integrate auto-notifications for delays or unpaid invoices, and ensure load times under 2 seconds with a responsive, user-friendly interface following industry standards (e.g., secure data handling).

Requirements

- **Delivery and Billing Integration Process**:
    - **Access**: Sales Executives/Managers (verified via RBAC) access delivery/billing data at /opportunities/edit/:id from L6 Won onward. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: "Access denied. Sales Executive/Manager role required."
    - **Interface Components**:
        - Delivery/Billing Section: <div class="delivery-billing"> below opportunity form in L6, displaying:
            - Delivery Progress: <select aria-label="Delivery Progress" disabled> (options: Pending, Partial, Complete, or %).
            - Billing Status: <select aria-label="Billing Status" disabled> (options: Not Billed, Partially Billed, Fully Billed).
            - Invoice Status: <select aria-label="Invoice Status" disabled> (options: Draft, Sent, Paid, Overdue).
            - Invoice Reference: <div aria-label="Invoice Reference"> (Number, Date, Amount, File Link, read-only).
        - Auto-Notifications: Display alerts (e.g., <p class="alert">Delivery delayed</p> or <p class="alert">Invoice overdue</p>).
        - Error message: Red text above section (<p class="error">...</p>).
        - Success message: Green text above section (<p class="success">Data synced successfully</p>).
        - Partial Delivery: <table class="delivery-items"> with item-level progress.
        - Auto-Notifications: <p class="alert">Delivery delayed</p>, <p class="alert">Invoice overdue</p>
    - **Data Sources**:
        - Sync read-only from ERP (e.g., SAP, Oracle, Dynamics, Tally) via API (/api/erp/sync/:opportunity_id).

        - Fields populated: Delivery Progress, Billing Status, Invoice Status, Invoice Number, Date, Amount, File Link.
        - Partial Delivery: SELECT item_id, delivery_status FROM order_items WHERE order_id = ?.
        - Currency: Convert amounts using exchange rates (SELECT rate FROM exchange_rates WHERE currency_code = ?).

    - **Integration Models**:
        - Model A (ERP-driven): Invoices generated in ERP/Finance, CRM displays details via integration.
        - Data Governance: Sales can view but not edit to keep financial data within Finance domain.
        - Payment Reconciliation: Match payments to invoices (SELECT invoice_id, payment_amount FROM payments WHERE invoice_id = ?). Show: "Payment discrepancy detected."
    - **Actions**:
        - Auto-notify Sales if Delivery Progress is delayed (>7 days from expected).
        - Auto-alert Sales if Invoice Status is Overdue (>30 days from Sent).
        - Support forecasting by exposing delivery/billing data for revenue recognition.
        - Enable proactive follow-ups for renewals/upsells when Delivery Progress is Complete and Billing Status is Fully Billed.
    - **Validation**:
        - Server-side (all validations performed by synced data):
            - Sync failure: Show error: "Failed to sync delivery/billing data from ERP."
            - Missing ERP data: Show error: "ERP data unavailable for this opportunity."
            - Delay detection: Trigger alert if delivery exceeds timeline, show error: "Delivery delayed beyond 7 days."
            - Overdue invoice: Trigger alert if unpaid >30 days, show error: "Invoice overdue."
            - Currency: Verify exists, show: "Invalid currency."
- **Interface and User Experience**:

    - **Layout**: Section below form (300px wide on desktop, 90% on mobile). Page background: light gray; section: white. Padding: 16px for fields.
    - **Colors**: Labels: gray; alerts: orange; success: green; error: red; data: black.
    - **Text**: Inter or Roboto font. Headings: 18px, bold; data: 14px (headings: 16px on mobile).
    - **Form Elements**: Disabled dropdowns and text, responsive layout. Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Align with Swayatta CRM logo, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Sync Failure: "Failed to sync delivery/billing data from ERP."
        - Missing ERP Data: "ERP data unavailable for this opportunity."
        - Delivery Delay: "Delivery delayed beyond 7 days."
        - Invoice Overdue: "Invoice overdue."
        - Unauthorized Access: "Access denied. Sales Executive/Manager role required."
- **Industry Standards**:
    - **Security**: Use HTTPS, encrypt synced data (e.g., invoice amounts), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log sync actions securely.
    - **Data Integrity**: Ensure read-only sync preserves ERP accuracy, prevent manual overrides.
    - **Integration**: Use RESTful API with OAuth for ERP sync, handle rate limits and retries.
- **Non-Functional Requirements**:
    - **Performance**: Load synced data and alerts in under 2 seconds. Cache ERP data, optimize API calls.
    - **Scalability**: Handle 10,000 opportunities with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry ERP sync 3 times (1s, 2s, 4s delays).
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log sync operations, delivery delays, and invoice alerts, track error rates.
- **Scenarios**:

    - Post-Win: Sync L6 Won opportunity, show Delivery Progress as Partial, Billing Status as Not Billed.
    - Delay Alert: Delivery Progress delayed, trigger auto-notification to Sales.
    - Overdue Invoice: Invoice Status as Overdue, alert Sales for follow-up.
    - Edge Cases: Sync failure (error shown), missing ERP data (error flagged), unauthorized access (blocked).
- **Technical Instructions**:

  Sub-Tasks
- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

8. Prompt for Marketing Campaign Management Module Development for Swayatta 4.0 CRM/ERP System

Develop a comprehensive marketing campaign management module for the Swayatta 4.0 CRM/ERP system, enabling Marketing Professionals to plan, track, and analyze campaigns by creating campaigns with type and budget, segmenting leads, executing and tracking performance, and calculating ROI. Include multi-channel support (e.g., email, events, social), lead segmentation automation, A/B testing, real-time analytics, and compliance with data privacy regulations (e.g., GDPR). Restrict access via role-based access control (RBAC), integrate with email/automation tools for execution, and provide alerts for budget overruns. Follow industry-standard CRM practices (e.g., secure data handling, responsive design), achieve load times under 2 seconds, and provide a clean, user-friendly, responsive interface inspired by Marketing Performance wireframes.

Requirements

- **Marketing Campaign Management Process**:
    - **Access Form**: Marketing Professionals (verified via RBAC) access /campaigns/add or /campaigns/edit/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Marketing Professional role required.”
    - **Interface Components**:
        - Form: <form> with inputs for:
            - Campaign ID: Auto-generated alphanumeric (display only).
            - Type: <select aria-label="Type"> (e.g., Email, Event, Social).
            - Budget: <input type="number" aria-label="Budget" min="0">.

    - Lead Segment: <select aria-label="Lead Segment" multiple> (valid lead IDs from database).
    - ROI: <input type="number" aria-label="ROI"> (calculated, optional manual override).
    - A/B Test Variants: <div class="ab-test"> with options for email templates (if Type=Email).
    - Schedule: <input type="date" aria-label="Campaign Start">, <select aria-label="Recurrence"> (One-Time, Daily, Weekly).
    - Submit: <button>Save Campaign</button>.
  - Performance View: <div class="performance"> with charts (e.g., ROI bar, engagement line).
  - Error message: Red text above form (<p class="error">...</p>).
  - Success message: Green text above form (<p class="success">Campaign saved successfully</p>).
- **Fields**:
  - Campaign ID: Auto-generated alphanumeric.
  - Type: Required, from predefined list (Email, Event, Social).
  - Budget: Required, positive numeric.
  - Lead Segment: Required, valid lead IDs (array).
  - ROI: Optional, numeric (calculated as (Revenue Generated - Budget) / Budget * 100).
  - Created On/By: Auto-filled with timestamp and Marketing Professional ID.
  - Modified On/By: Auto-filled on updates.
  - Budget Currency: Required, from currencies table.
  - Schedule: Required, date and recurrence.
- **Validation**:
  - Server-side (all validations performed by Marketing Professional input):
    - Budget: Check positive numeric (/^\d+(\.\d{1,2})?$/), show error: “Budget must be positive.”
    - Lead Segment: Verify valid IDs (SELECT id FROM leads WHERE id IN (?)), show error: “Invalid lead IDs in segment.”
    - Type: Ensure from list, show error: “Invalid campaign type.”

- Budget overrun: Monitor spent vs. budget in real-time, show error: “Budget overrun detected.”
- Database failure: Show error: “Failed to save campaign. Try again.”
- Currency: Verify exists, show: “Invalid currency.”
- Schedule: Not past, show: “Campaign start date cannot be in the past.”

- Campaign Scheduling:
  - Schedule campaigns with recurrence. Store in campaign_schedules table. Show: “Invalid schedule.”
- Cross-Channel Attribution:
  - Track conversions (SELECT channel, conversion_count FROM campaign_metrics WHERE campaign_id = ?). Display multi-channel chart.
    - **Campaign Execution and Tracking**:
      - Segment leads automatically (e.g., by demographics, behavior from database).
      - Execute: Integrate with email tools (e.g., SendGrid) for Email, track engagement (opens, clicks).
      - Analyze: Calculate ROI in real-time, support A/B testing (variant performance comparison).
      - Notifications: Alert on budget overrun or low engagement.
    - **Privacy Compliance**:
      - Ensure lead segmentation complies with GDPR (consent checks), show error: “Lead segment violates privacy regulations.”
  - **Interface and User Experience**:
    - **Layout**: Center a card (400px wide on desktop, 90% on mobile). Page background: light gray; card: white. Padding: 32px for card, 16px for fields.
    - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
    - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
    - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Performance charts with tooltips. Error/success messages above form. Submit button at bottom.
    - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).

    - **Branding**: Add Swayatta CRM logo above form, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Invalid Budget: “Budget must be positive.”
        - Invalid Lead Segment: “Invalid lead IDs in segment.”
        - Invalid Type: “Invalid campaign type.”
        - Budget Overrun: “Budget overrun detected.”
        - Privacy Violation: “Lead segment violates privacy regulations.”
        - Database Failure: “Failed to save campaign. Try again.”
        - Unauthorized Access: “Access denied. Marketing Professional role required.”
- **Industry Standards**:
    - **Security**: Use HTTPS, encrypt sensitive data (e.g., lead IDs), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
    - **Data Integrity**: Enforce lead consent for segmentation, prevent deletion of active campaigns.
    - **Integration**: Support API with email/automation tools (e.g., SendGrid, Mailchimp), real-time tracking with analytics (e.g., Google Analytics).
    - **Compliance**: GDPR/CCPA for data privacy, opt-out mechanisms for leads.
- **Non-Functional Requirements**:
    - **Performance**: Load form and calculate ROI in under 2 seconds. Cache lead data, optimize queries.
    - **Scalability**: Handle 10,000 campaigns with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry executions 3 times (1s, 2s, 4s delays).
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log CRUD operations and errors, track campaign engagement and ROI metrics.
- **Scenarios**:
    - Campaign Creation: Marketing Professional creates Email campaign, segments leads, executes.
    - ROI Calculation: Track engagement, calculate ROI, alert on low performance.

    - Budget Overrun: Spend exceeds budget, trigger alert.
    - Edge Cases: Invalid segment (error shown), privacy violation (error flagged), execution failure (retry, then error).
- **Technical Instructions**:

  Sub-Tasks
- Adding of Technical instructions and Generation of Code
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

9. **Prompt for SLA/TAT Monitoring, Notification, and Checklist Module Development for Swayatta 4.0 CRM/ERP System**

Develop an integrated module for SLA/TAT monitoring, notification process, and Go/No-Go checklist enforcement in the Swayatta 4.0 CRM/ERP system, enabling system-driven tracking of SLAs inherited from orders, real-time monitoring with escalations and notifications, and mandatory checklists for entity progression (company, lead, opportunity). Actors include System, Managers, and Admins. Include multi-channel notifications (Email, SMS, In-app), compliance with regulations (e.g., GDPR for notifications), automation for breach detection, and audit trails for all actions. Restrict access via role-based access control (RBAC), provide real-time alerts, and ensure load times under 2 seconds with a responsive, user-friendly interface inspired by Monitoring Dashboard and Checklist Forms wireframes.

Requirements

- **SLA/TAT Monitoring Process**:
    - **Access Dashboard**: Managers (verified via RBAC) access /sla/monitor. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: "Access denied. Manager role required."
    - **Interface Components**:
        - Dashboard: <div class="sla-dashboard"> with table displaying SLA ID, Name, Status, Response/Resolution Time, and breach alerts.
        - Error message: Red text above dashboard (<p class="error">...</p>).
        - Success message: Green text above dashboard (<p class="success">SLA updated successfully</p>).
        - Benchmarking Chart
    - **Fields**:
        - SLA ID: Alphanumeric, 6-20 characters.
        - Name: Required, 3-100 characters.

        - Customer Tier/Service Type/Priority: Required, from predefined dropdowns (e.g., Gold/Silver, High/Medium/Low).
        - Response/Resolution Time: Required, HH:MM (>=0).
        - Business Hours/Days: Required, predefined (e.g., 9-5 Mon-Fri).
        - Start/End Date: Required, date range.
        - Regions: Required, dropdown (e.g., APAC, EMEA).
        - Escalation Matrix: Required, JSON array of levels/users.
        - Status: Required, Active/Inactive.
        - Auto-Renew: Required, boolean.
        - Penalty/Comments: Optional, up to 500 characters.
    - **Validation**:
        - Server-side:
            - Resolution Time: Check >= Response Time, show error: “Resolution time must be greater than or equal to response time.”
            - SLA ID: Check format (/^[a-zA-Z0-9]{6,20}$/), show error: “SLA ID must be 6-20 alphanumeric characters.”
            - Mandatory fields: Prevent submission if empty, show error: “All required fields must be filled.”
            - Breach detection: Monitor real-time, escalate if breached, show error: “SLA breach detected. Escalated.”
            - Database failure: Show error: “Failed to save SLA. Try again.”
    - **Monitoring and Escalation**:
        - Inherit SLA from order (SELECT sla_id FROM orders WHERE order_id = ?).
        - Monitor real-time using cron jobs/timers, notify on breach via escalation matrix.
        - Exceptions: Unavailability -> Reassign task, show error: “Resource unavailable. Reassigned.”
- **Notification Process**:
    - **Detection and Sending**: System detects events (e.g., breach, approval needed), selects channel, sends/logs notification.
    - **Fields**:
        - Event ID: Auto-generated.

        - Channel: Required, from list (Email, SMS, In-app).
        - Recipient: Required, user ID.
    - **Validation**:
        - Server-side:
            - Channel: Check valid, show error: "Invalid notification channel."
            - Recipient: Verify user ID (SELECT id FROM users WHERE id = ?), show error: "Invalid recipient."
            - Failure: Retry 3 times, show error: "Failed to send notification. Retried."
    - **Multi-Channel Support**: Integrate with email/SMS APIs (e.g., SendGrid, Twilio) and in-app (WebSocket for real-time).
- **Checklist Process**:
    - **Access Form**: Admins (verified via RBAC) access /checklists/add or /checklists/edit/:id. Check role server-side. If unauthorized, show error: "Access denied. Admin role required."
    - Customizable Checklists

- SLA Benchmarking:
    - Compare metrics (SELECT response_time, resolution_time FROM slas) against industry standards. Display chart.
        - **Fields**:
            - Checklist ID: Auto-generated.
            - Entity ID: Required, valid ID (company/lead/opportunity).
            - Step: Required, up to 100 characters.
            - Status: Required, Pending/Completed.
            - Description: Optional, up to 500 characters.
            - Started/Completed On/By: Auto-filled timestamps and user IDs.
            - Options: Optional, JSON array.
        - **Validation**:
            - Server-side:
                - Incomplete: Block progression if not all completed, show error: "Checklist incomplete. Cannot proceed."
                - Entity ID: Verify existence (SELECT id FROM entities WHERE id = ?), show error: "Invalid entity ID."

  - **Enforcement**: Link to entities, validate before progression, approve on completion.

- **Integration**:
  - Notifications for SLA breaches/checklist completions.
  - Audit trails for all processes.
- **Interface and User Experience**:
  - **Layout**: Dashboard with sections for SLA monitoring, notifications, checklists (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 32px for cards, 16px for fields.
  - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
  - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
  - **Form Elements**: Full-width inputs/dropdowns, rounded corners. In-app notifications as pop-ups. Error/success messages above forms. Submit button at bottom.
  - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
  - **Branding**: Add Swayatta CRM logo above forms, keep design clean and professional.
- **Error Messages and Validation**:
  - **Server-Side**:
    - Invalid SLA ID: “SLA ID must be 6-20 alphanumeric characters.”
    - Resolution < Response: “Resolution time must be greater than or equal to response time.”
    - SLA Breach: “SLA breach detected. Escalated.”
    - Invalid Channel: “Invalid notification channel.”
    - Invalid Recipient: “Invalid recipient.”
    - Notification Failure: “Failed to send notification. Retried.”
    - Incomplete Checklist: “Checklist incomplete. Cannot proceed.”
    - Invalid Entity ID: “Invalid entity ID.”
    - Unauthorized Access: “Access denied. Required role not found.”
    - Database Failure: “Failed to process. Try again.”
- **Industry Standards**:

    - **Security**: Use HTTPS, encrypt sensitive data (e.g., escalation matrix), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
    - **Data Integrity**: Enforce SLA inheritance, prevent deletion of active SLAs/checklists.
    - **Compliance**: GDPR for notifications (consent checks), audit trails for regulatory reporting.
    - **Integration**: WebSockets for real-time monitoring/notifications, multi-channel APIs (e.g., Twilio for SMS).
- **Non-Functional Requirements**:
    - **Performance**: Load dashboards and process escalations in under 2 seconds. Cache predefined lists, optimize queries.
    - **Scalability**: Handle 10,000 entities with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry notifications/escalations 3 times (1s, 2s, 4s delays).
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log events and errors, track SLA breaches, notification deliveries, and checklist completions.
- **Scenarios**:
    - SLA Breach: Monitor SLA, detect breach, escalate and notify.
    - Notification: Detect event (e.g., approval needed), send via Email/SMS/In-app.
    - Checklist: Complete steps for opportunity, validate, approve progression.
    - Edge Cases: Budget overrun in monitoring (alert shown), notification failure (retry, then error), incomplete checklist (block progression).
- **Technical Instructions**:

Sub-Tasks

- Adding of Technical functionality
- Deployment
- Fixing of Bugs Manually
- Testing and Monitoring

10. **Prompt for Change Account Owner and Conflicts Management Module Development for Swayatta 4.0 CRM/ERP System**

Develop a module for Change Account Owner and Conflicts Management in the Swayatta 4.0 CRM/ERP system, enabling Admins, Managers, Sales Heads, and Executive Committee (EC) members to reassign ownership of entities (e.g., leads, opportunities, accounts) and resolve conflicts efficiently. The process includes selecting an entity, choosing a new owner with validated permissions, updating ownership, and notifying stakeholders. For conflicts, detect overlapping assignments or AI-detected issues, notify relevant parties, escalate as needed, and resolve or update records. Ensure role-based access control (RBAC), audit trails for ownership changes, and compliance with data privacy standards (e.g., GDPR). Achieve load times under 2 seconds with a responsive, user-friendly interface inspired by reassign/escalate forms.

Requirements

- **Change Account Ownership Process**:
    - **Access Form**: Admins/Managers/Sales Heads/EC (verified via RBAC) access /ownership/change/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Required role not found.”
    - **Interface Components**:
        - Form: <form> with inputs:
            - Entity ID: <input type="text" aria-label="Entity ID" readonly> (valid entity ID).
            - New Owner: <select aria-label="New Owner"> (valid user IDs from database).
            - Entities: Array of valid entity IDs for bulk reassignment.
            - Submit: <button>Reassign Ownership</button>.
        - Notification: <p class="notification">Ownership updated for [Entity ID]</p> (green text).
        - Error message: Red text above form (<p class="error">...</p>).
    - **Fields**:
        - Entity ID: Required, valid entity ID (e.g., lead, opportunity).
        - New Owner: Required, valid user ID with appropriate role.
    - **Validation**:
        - Server-side:
            - Entity ID: Verify existence (SELECT id FROM entities WHERE id = ?), show error: “Invalid entity ID.”
            - New Owner: Check permissions (SELECT role FROM users WHERE id = ?), show error: “New owner lacks required permissions.”

        - Unauthorized access: Deny if role mismatch, show error: “Access denied. Required role not found.”
        - Database failure: Show error: “Failed to update ownership. Try again.”
    - **Steps**: Select entity, choose new owner, validate permissions, update ownership, notify old/new owners via email/in-app.
- **Conflicts Management Process**:
    - **Access Form**: Admins/Managers/Sales Heads/EC access /conflicts/resolve/:id. Check role server-side.
    - **Interface Components**:
        - Form: <form> with inputs:
            - Conflict ID: <input type="text" aria-label="Conflict ID" readonly> (auto-generated).
            - Entity IDs: <input type="text" aria-label="Entity IDs" readonly> (valid IDs involved).
            - Resolution: <textarea aria-label="Resolution" maxlength="500"></textarea> (optional).
            - Submit: <button>Resolve Conflict</button>.

Conflict History: <div class="conflict-history"> with past resolutions.

Validation: Entity IDs (SELECT id FROM entities WHERE id IN (?)), show: “Invalid entity IDs.” [Error Code: OWN_006]
Resolution: Limit 500 chars, show: “Resolution exceeds 500 characters.”

Bulk Ownership Changes:

- Reassign multiple entities. Show: “Bulk reassignment completed.”

Conflict Resolution History:

- Display logs (SELECT conflict_id, resolution, timestamp FROM audit_logs WHERE entity_type = 'conflict').
        - Escalation Alert: <p class="alert">Conflict escalated to [role]</p> (orange text).
        - Error message: Red text above form (<p class="error">...</p>).
    - **Fields**:
        - Conflict ID: Auto-generated alphanumeric.

        - Entity IDs: Required, valid IDs of conflicting entities.
        - Resolution: Optional, up to 500 characters.
    - **Validation**:
        - Server-side:
            - Entity IDs: Verify existence (SELECT id FROM entities WHERE id IN (?)), show error: "Invalid entity IDs."
            - Unauthorized access: Deny if role mismatch, show error: "Access denied. Required role not found."
            - Resolution: Limit to 500 chars, show error: "Resolution exceeds 500 characters."
            - Database failure: Show error: "Failed to resolve conflict. Try again."
    - **Steps**: Detect conflict (e.g., AI identifies overlapping ownership), notify stakeholders, escalate to higher role if unresolved, resolve/update records.
    - **Conflict Detection**: Use AI to flag duplicate assignments or ownership disputes, trigger escalation if unresolved within 24 hours.
- **Integration**:
    - Notify stakeholders via multi-channel support (Email, In-app) for ownership changes and conflict resolutions.
    - Maintain audit logs for all actions.
- **Interface and User Experience**:
    - **Layout**: Forms with cards (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 32px for cards, 16px for fields.
    - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; alerts: orange; labels: gray; inputs: white with gray border.
    - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
    - **Form Elements**: Full-width inputs/dropdowns, rounded corners. In-app notifications as pop-ups. Error/success messages above forms. Submit button at bottom.
    - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Add Swayatta CRM logo above forms, keep design clean and professional.
- **Error Messages and Validation**:

    - **Server-Side**:
        - Invalid Entity ID: “Invalid entity ID.”
        - Permission Denied: “New owner lacks required permissions.”
        - Unauthorized Access: “Access denied. Required role not found.”
        - Invalid Entity IDs: “Invalid entity IDs.”
        - Resolution Limit: “Resolution exceeds 500 characters.”
        - Ownership Update Failure: “Failed to update ownership. Try again.”
        - Conflict Resolution Failure: “Failed to resolve conflict. Try again.”
- **Industry Standards**:
    - **Security**: Use HTTPS, encrypt sensitive data (e.g., user IDs), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
    - **Data Integrity**: Prevent duplicate ownership, enforce permission checks, maintain historical records.
    - **Compliance**: GDPR for notifications (consent checks), audit trails for regulatory reporting.
    - **Integration**: WebSockets for real-time conflict detection, email/SMS APIs (e.g., SendGrid, Twilio).
- **Non-Functional Requirements**:
    - **Performance**: Load forms and process updates in under 2 seconds. Cache user/role data, optimize queries.
    - **Scalability**: Handle 10,000 entities with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry notifications 3 times (1s, 2s, 4s delays).
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log ownership changes, conflict escalations, and notification deliveries, track error rates.
- **Scenarios**:
    - Ownership Change: Reassign lead to new owner, notify stakeholders.
    - Conflict Resolution: Detect overlap, escalate to EC, resolve with comments.
    - Edge Cases: Unauthorized access (denied), conflict unresolved (further escalate), sync failure (retry, then error).
- **Technical Instructions**:

Sub-Tasks

- Adding of technical functionalities and generate code
- Deployment
- Fixing of Bugs Manually
- Testing and monitoring

11. **Prompt for Channel Partner Management and EC Review Module Development for Swayatta 4.0 CRM/ERP System**

    Develop a module for Channel Partner Management and EC Review in the Swayatta 4.0 CRM/ERP system, enabling Partner Managers to onboard partners by registering details, assigning SKUs and commissions, integrating sales data, and tracking performance, while EC Members conduct quarterly reviews by scheduling, collecting data, voting, and recording decisions. Ensure role-based access control (RBAC), compliance with data privacy standards (e.g., GDPR), automated sales integration, and audit trails for all actions. Support multi-tier partner hierarchies and real-time performance tracking. Achieve load times under 2 seconds with a responsive, user-friendly interface inspired by Partner Forms and Review Voting wireframes.

    Requirements

    - **Channel Partner Management Process**:
        - **Access Form**: Partner Managers (verified via RBAC) access /partners/add or /partners/edit/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Partner Manager role required.”
        - **Interface Components**:
            - Form: <form> with inputs:
                - Partner Name: <input type="text" aria-label="Partner Name" maxlength="255">.
                - SKU: <select aria-label="SKU" multiple> (catalog items).
                - Commission Rate: <input type="number" aria-label="Commission Rate" min="0" step="0.01">.
                - Submit: <button>Register Partner</button>.
            - Tracking Dashboard: <div class="partner-dashboard"> with sales performance metrics.
            - Partner Portal: <div class="partner-portal"> at /partners/portal.
            - Error message: Red text above form (<p class="error">...</p>).
            - Success message: Green text above form (<p class="success">Partner registered successfully</p>).

- **Fields**:
  - Partner Name: Required, up to 255 characters.
  - SKU: Required, from catalog items.
  - Commission Rate: Required, positive numeric.
  - Integration Status: Auto-filled (Pending/Integrated).
  - Created On/By: Auto-filled with timestamp and Partner Manager ID.
  - Commission Currency: Required, from currencies table.
- **Validation**:
  - Server-side:
    - Partner Name: Check length (/^.{%3,255}$/), show error: “Partner name must be 3-255 characters.”
    - SKU: Verify catalog existence (SELECT id FROM skus WHERE id IN (?)), show error: “Invalid SKU selected.”
    - Commission Rate: Check positive numeric (/^\d+(\.\d{1,2})?$/), show error: “Commission rate must be positive.”
    - Missing data: Postpone integration, show error: “Missing data. Integration postponed.”

- Partner Portal:
  - Allow partners to view metrics (SELECT sales_volume, commission FROM partner_metrics WHERE partner_id = ?). RBAC enforced.
- Incentive Programs:
  - Define tiered commissions/bonuses (<input type="number" aria-label="Bonus Rate">). Track payouts (INSERT INTO partner_payouts).

    - Database failure: Show error: “Failed to register partner. Try again.”
    - Currency: Verify exists, show: “Invalid currency.”

- **Steps**: Register details, assign SKUs/commissions, integrate sales data (via API), track performance (sales volume, commission earned).
- **Integration**: Sync sales data from partner systems, automate commission calculations.

- **EC Review Process**:
  - **Access Form**: EC Members (verified via RBAC) access /reviews/schedule or /reviews/vote/:id. Check role server-side. If unauthorized, show error: “Access denied. EC Member role required.”

- **Interface Components**:
    - Form: <form> with inputs:
        - Review ID: <input type="text" aria-label="Review ID" readonly> (auto-generated).
        - Vote: <select aria-label="Vote"> (Approve/Reject).
        - Decision: <textarea aria-label="Decision" maxlength="500"></textarea> (optional).
        - Submit: <button>Record Decision</button>.
    - Schedule Section: <div class="schedule"> with date picker and data collection fields.
    - Error message: Red text above form (<p class="error">...</p>).
    - Success message: Green text above form (<p class="success">Review recorded successfully</p>).
- **Fields**:
    - Review ID: Auto-generated alphanumeric.
    - Vote: Required, Approve/Reject.
    - Decision: Optional, up to 500 characters.
    - Review Date: Required, date picker.
    - Data Collected: Auto-filled from partner performance.
- **Validation**:
    - Server-side:
        - Missing data: Postpone review, show error: “Missing data. Review postponed.”
        - Vote: Check valid option, show error: “Invalid vote selection.”
        - Decision: Limit to 500 chars, show error: “Decision exceeds 500 characters.”
        - Unauthorized access: Deny if role mismatch, show error: “Access denied. EC Member role required.”
        - Database failure: Show error: “Failed to record review. Try again.”
- **Steps**: Schedule review, collect partner performance data, vote, record decision.
- **Scheduling**: Automate quarterly reviews, notify EC Members.

- **Integration**:
  - Notify Partner Managers and EC Members via multi-channel support (Email, In-app) for onboarding and reviews.
  - Maintain audit logs for all actions.
- **Interface and User Experience**:
  - **Layout**: Forms with cards (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 32px for cards, 16px for fields.
  - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
  - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
  - **Form Elements**: Full-width inputs/dropdowns, rounded corners. In-app notifications as pop-ups. Error/success messages above forms. Submit button at bottom.
  - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
  - **Branding**: Add Swayatta CRM logo above forms, keep design clean and professional.
- **Error Messages and Validation**:
  - **Server-Side**:
    - Invalid Partner Name: “Partner name must be 3-255 characters.”
    - Invalid SKU: “Invalid SKU selected.”
    - Invalid Commission Rate: “Commission rate must be positive.”
    - Missing Data (Partners): “Missing data. Integration postponed.”
    - Missing Data (EC): “Missing data. Review postponed.”
    - Invalid Vote: “Invalid vote selection.”
    - Decision Limit: “Decision exceeds 500 characters.”
    - Unauthorized Access: “Access denied. Required role not found.”
    - Database Failure: “Failed to process. Try again.”
- **Industry Standards**:
  - **Security**: Use HTTPS, encrypt sensitive data (e.g., partner details), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
  - **Data Integrity**: Enforce unique partner names, prevent deletion of active partners/reviews.

    - **Compliance**: GDPR for partner data (consent checks), audit trails for regulatory reporting.
    - **Integration**: RESTful APIs for sales integration, multi-channel notifications (e.g., SendGrid for Email).
- **Non-Functional Requirements**:
    - **Performance**: Load forms and process updates in under 2 seconds. Cache SKU data, optimize queries.
    - **Scalability**: Handle 10,000 partners with efficient indexing and caching.
    - **Reliability**: Ensure 99.9% uptime, retry notifications 3 times (1s, 2s, 4s delays).
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log onboarding, review actions, and notification deliveries, track error rates.
- **Scenarios**:
    - Partner Onboarding: Register partner, assign SKUs, integrate sales, track performance.
    - EC Review: Schedule quarterly review, vote Approve, record decision.
    - Edge Cases: Missing data (postponed), unauthorized access (denied), sync failure (retry, then error).
- **Technical Instructions**:

Sub-Tasks

- Adding of technical functionalities and generate code
- Deployment
- Fixing of Bugs Manually
- Testing and monitoring

12. **Prompt for ERP Integrations and Non-Functional Requirements Module Development for Swayatta 4.0 CRM/ERP System**

Develop a module for ERP integrations (Order-to-Cash [O2C] and Procure-to-Pay [P2P]) and enforce non-functional requirements across the Swayatta 4.0 CRM/ERP system, enabling seamless syncing of data between CRM and ERP systems using stable APIs. For O2C, sync won opportunities to orders, invoices, and BOMs, while tracking payments and revenue. For P2P, onboard partners/vendors, procure SKUs, and trigger commissions/expenses. Incorporate these integrations into all relevant modules (e.g., opportunities, partners), using a monolithic design with SQL database and cloud deployment (e.g., AWS/Azure). Enforce system-wide non-functional requirements for performance, security, reliability, usability, scalability, and compliance (GDPR/PCI-DSS). Include integration dashboards for monitoring, automated error handling, and logging for audits. Follow industry-standard practices (e.g., API rate limiting, circuit breakers, containerization for deployment).

Requirements

- **ERP Integrations Process**:
  - **O2C Integration**: Sync won opportunities from CRM to ERP orders (/api/o2c/sync/:opportunity_id), generate invoices/BOMs, track payments/revenue in real-time. Use stable APIs (e.g., RESTful with OAuth), map fields like opportunity ID to order ID.
  - **P2P Integration**: Sync partner/vendor onboarding to ERP procurement (/api/p2p/sync/:partner_id), procure SKUs via API calls, trigger commissions/expenses on sales milestones.
  - Bidirectional Sync: Update CRM from ERP (/api/o2c/update/:order_id). Show: “Bidirectional sync failed.”
  - **Access Dashboard**: Admins/Managers (verified via RBAC) access /integrations/dashboard. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: “Access denied. Required role not found.”
  - **Interface Components**:
    - Dashboard: <div class="integration-dashboard"> with tables for sync status, logs, and metrics (e.g., success rate, failures).
    - Sync Button: <button>Manual Sync</button> for O2C/P2P.
    - Error Recovery Queue: <div class="sync-queue"> for failed syncs.
    - Error message: Red text above dashboard (<p class="error">...</p>).
    - Success message: Green text above dashboard (<p class="success">Sync completed successfully</p>).
  - **Validation**:
    - Server-side:
      - Sync Data: Verify data integrity (e.g., opportunity status = Won), show error: “Invalid data for sync.”
      - API Failure: Retry 3 times, show error: “API sync failed. Retried.”
      - Unauthorized access: Deny if role mismatch, show error: “Access denied. Required role not found.”
      - Database failure: Show error: “Failed to sync data. Try again.”
- **Non-Functional Enforcement**:

  - **Performance**: Ensure <2s response times for all operations, support 1000 concurrent users with load balancing.
  - **Security**: Implement role-based access, AES-256 encryption for sensitive data, comprehensive logging for audits.
  - **Reliability**: Achieve 99.9% uptime with daily backups, redundancy, and failover mechanisms.
  - **Usability**: Consistent CSS across modules, responsive design for web/mobile.
  - **Scalability**: Horizontal scaling, auto-scaling on cloud.
  - **Compliance**: GDPR/PCI-DSS for data handling, consent management for integrations.
  - Incorporate into all modules: Add sync hooks in opportunities/partners, enforce non-functionals system-wide.
- **Interface and User Experience**:
  - **Layout**: Dashboard with tabs for O2C/P2P (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 32px for cards, 16px for fields.
  - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
  - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
  - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Error/success messages above forms. Submit button at bottom.
  - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
  - **Branding**: Add Swayatta CRM logo above dashboard, keep design clean and professional.
- **Error Messages and Validation**:
  - **Server-Side**:
    - Invalid Sync Data: “Invalid data for sync.”
    - API Failure: “API sync failed. Retried.”
    - Unauthorized Access: “Access denied. Required role not found.”
    - Database Failure: “Failed to sync data. Try again.”
    - Performance Issue: “Response time exceeded 2s. Check system load.”
    - Currency: Verify exists, show: “Invalid currency.”

        - Error Recovery Workflows: Queue failed syncs for review (INSERT INTO sync_queue). Notify Admins, show: “Sync queued for review.”

- **Industry Standards**:
    - **Security**: Use HTTPS, OAuth for APIs, encryption for PII, logging with ELK stack.
    - **Data Integrity**: Transactional syncs with ACID compliance, rollback on failures.
    - **Compliance**: GDPR for data transfers, PCI-DSS for payment tracking.
    - **Integration**: Use stable APIs with rate limiting, circuit breakers for resilience.
- **Non-Functional Requirements**:
    - **Performance**: <2s response, 1000 users with load testing.
    - **Security**: Role-access, encryption, logs.
    - **Reliability**: 99.9% uptime, backups.
    - **Usability**: Consistent CSS, responsive.
    - **Scalability**: Cloud deploy with auto-scaling.
    - **Compliance**: GDPR/PCI.
- **Scenarios**:
    - Opportunity to O2C Sync: Won opportunity synced to ERP order/invoice.
    - Vendor Payments: P2P sync triggers commission payment.
    - Edge Cases: Sync failure (retry, then error), unauthorized access (denied).
- **Technical Instructions**:

Sub-Tasks

- Adding of technical functionalities and generate code
- Deployment
- Fixing of Bugs Manually
- Testing and monitoring

13. **Prompt for AI/LLM Features Module Development for Swayatta 4.0 CRM/ERP System**

Develop an AI/LLM features module for the Swayatta 4.0 CRM/ERP system, enabling system-driven processing of data using large language models (LLMs) for deduplication, report generation, and Purchase Order (PO) analysis. The module inputs data (text/image), processes via LLM, outputs insights with confidence scores, and allows user review/override for low-confidence results. Integrate with models like GPT-4 or fine-tuned variants, ensure bias mitigation, and comply with data privacy (e.g., GDPR). Provide an insights dashboard for visualization, automate workflows for high-confidence outputs, and include logging for AI decisions. Follow industry-standard AI practices (e.g., explainable AI, ethical guidelines), achieve processing times under 5 seconds, and provide a clean, user-friendly, responsive interface inspired by Insights Dashboard wireframes.

Requirements

- **AI/LLM Processing Process**:
  - **Access Dashboard**: Users (verified via RBAC) access /ai/insights. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: "Access denied. Required role not found."
  - **Interface Components**:
    - Form: <form> with inputs:
      - Input Data: <input type="file" aria-label="Input Data" accept="text/plain,image/*"> (text/image).
      - Output Type: <select aria-label="Output Type"> (e.g., Duplicate Detection, Report Generation, PO Extraction).
      - Submit: <button>Process Data</button>.
    - Insights Dashboard: <div class="insights-dashboard"> with table/charts for outputs, confidence scores, and review options.
    - Review: <button>Override</button> for low-confidence results.
    - Error message: Red text above form (<p class="error">...</p>).
    - Success message: Green text above form (<p class="success">Insights generated successfully</p>).
  - **Fields**:
    - Input Data: Required, text or image.
    - Output Type: Required, from predefined list (Duplicate, Report, PO Analysis).
    - Confidence: Auto-generated, 0-100%.
    - Processed On/By: Auto-filled with timestamp and system/user ID.
  - **Validation**:
    - Server-side:
      - Input Data: Check format/size (text <1MB, image <5MB), show error: "Invalid input data format or size."
      - Output Type: Ensure from list, show error: "Invalid output type."
      - Low Confidence: If <70%, route to human review, show error: "Low confidence. Human review required."

- Database failure: Show error: “Failed to process data. Try again.”

- **Steps**: Input data, process with LLM (e.g., dedupe contacts via similarity matching, extract PO details via OCR/LLM parsing, generate reports via summarization), output insights, auto-save high-confidence results, review/override low-confidence.
- **Exceptions**: Low confidence -> Route to user review queue.

- **Interface and User Experience**:
  - **Layout**: Dashboard with upload section and results table (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 32px for cards, 16px for fields.
  - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
  - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
  - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Confidence as progress bar. Error/success messages above form. Submit button at bottom.
  - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
  - **Branding**: Add Swayatta CRM logo above dashboard, keep design clean and professional.
- **Error Messages and Validation**:
  - **Server-Side**:
    - Invalid Input Data: “Invalid input data format or size.”
    - Invalid Output Type: “Invalid output type.”
    - Low Confidence: “Low confidence. Human review required.”
    - Database Failure: “Failed to process data. Try again.”
    - Unauthorized Access: “Access denied. Required role not found.”
- **Industry Standards**:
  - **Security**: Use HTTPS, encrypt input data, parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely (no sensitive data).
  - **AI Ethics**: Mitigate bias in LLM outputs, provide explainability (e.g., confidence rationale), comply with GDPR for data processing.
  - **Integration**: Use LLMs like GPT-4 via APIs (e.g., OpenAI, with LangChain for chaining), OCR for images (e.g., Tesseract).

    - **Compliance**: Ensure AI decisions are auditable, opt-in for AI processing.
- **Non-Functional Requirements**:
    - **Performance**: Process data in under 5 seconds. Cache LLM responses, optimize API calls.
    - **Scalability**: Handle 10,000 requests/day with efficient queuing.
    - **Reliability**: Ensure 99.9% uptime, retry LLM calls 3 times (1s, 2s, 4s delays).
    - **Maintainability**: Write modular code with documentation, use formatting tools, store secrets in environment variables.
    - **Monitoring**: Log AI outputs and errors, track confidence scores and review rates.
- **Scenarios**:
    - Dedupe Contacts: Input contact list, detect duplicates with 95% confidence, auto-merge.
    - PO Extract: Input PO image, extract details with 60% confidence, route to review.
    - Edge Cases: Invalid input (error shown), low confidence (review triggered), LLM failure (retry, then error).
- **Technical Instructions**:

Sub-Tasks

- Adding of technical functionalities and generate code
- Deployment
- Fixing of Bugs Manually
- Testing and monitoring

14. Prompt for Deal Governance Module Development for Swayatta 4.0 CRM/ERP System

Develop a deal governance module for the Swayatta 4.0 CRM/ERP system to enforce a standardized framework for procurement and deal management across the lifecycle—from RFP qualification to closure—enhancing cross-functional accountability, legal/financial compliance, and profitability. Integrate with the sales CRM to manage opportunities, incorporating the ETVOX framework (Entry Task Verification Output Exit) for task timelines, stakeholder sign-offs at predefined checkpoints, and automated deal tracking. Apply to deals ≥ ₹1 Cr, strategic/complex bids, third-party procurement, and formal contracts. Ensure role-based access control (RBAC), audit trails with sign-off timestamps and approver IDs, and compliance with GDPR/PCI-DSS, achieving load times under 2 seconds with a responsive, user-friendly interface inspired by deal tracking dashboards.

Requirements

- **Deal Governance Process**:
    - **Access Dashboard**: Sales, Pre-Sales, Delivery, Legal, Finance, Procurement, COE/PMO, and Client Manager (verified via RBAC) access /deals/governance/:id. Check role server-side (SELECT role FROM users WHERE user_id = ?). If unauthorized, show error: "Access denied. Required role not found."
    - **Interface Components**:
        - Dashboard: <div class="deal-dashboard"> with stages (Opportunity Qualification, Solution Design, Commercial & Legal Review, HOTO, Procurement Execution, Change Management), task timelines, sign-off statuses, and Client Manager updates.
        - Form: <form> per stage with dynamic inputs (e.g., RFP details, SoW, PO mapping, Change Impact Template).
        - Submit/Sign-off: <button>Submit/Sign-off</button> per checkpoint.
        - Error message: Red text above dashboard (<p class="error">...</p>).
        - Success message: Green text above dashboard (<p class="success">Task completed successfully</p>).
    - **Fields** (Stage-Specific):
        - Opportunity Qualification: Deal value, complexity, stakeholder DL.
        - Solution Design: Solution design document, Delivery sign-off, BoQ/BOM draft.
        - Commercial & Legal Review: Legal sign-off sheet, Finance CBA.
        - HOTO: SoW, cost model, PO mapping validation.
        - Procurement Execution: Indent form, POT-ID, purchase order register.
        - Change Management: Change request log, Change Impact Template (cost, timeline, legal, vendor), escalation details.
    - **Validation**:
        - Server-side:
            - Mandatory Sign-offs: Block progression if missing, show error: "Required sign-off missing."
            - TAT Breach: Alert if task exceeds timeline (e.g., N-5 for RFP), show error: "Task overdue."
            - Budget Deviation: Flag >5% overrun, show error: "Budget deviation detected."
            - Change Impact: Escalate if >10% delta, show error: "Change exceeds 10% threshold."
            - POT-ID/BoQ Validation: Verify against ERP, show error: "Invalid POT-ID or BoQ."
            - Database failure: Show error: "Failed to update deal. Try again."
    - **Stages and Actions**:
        - Opportunity Qualification (Sales, Pre-Sales): Evaluate deal, trigger stakeholder DL (N-5).
        - Solution Design & Pre-Bid (Pre-Sales, Delivery): Design solution, get Delivery sign-off, draft BoQ/BOM (N-4 to N-2).
        - Commercial & Legal Review (Legal, Finance): Legal compliance, Finance CBA (N-2).

        - Handover Post-Win (HOTO) (Pre-Sales, Finance): Transfer SoW/cost model, validate PO (N+1).
        - Procurement Execution (Delivery, Procurement): Raise indent with POT-ID, execute purchase (N+2 to N+7).
        - Change Management (Delivery, Finance, Legal): Assess impact via Change Impact Template, escalate to EC/PMO if >10% (ongoing).
    - **Automation**: Integrate deal tracker with procurement, indent, cost validation, and ERP systems. Automate RACI-based notifications.
- **Process Controls & Governance**:
    - Validation & Audits: COE/PMO conducts audits, tracks sign-offs with timestamps and approver IDs.
    - Metrics (KPIs): % timely Legal/Finance involvement, budget deviation, SLA adherence, escalated CRs.
    - Exceptions: Exemptions for strategic/emergency cases require Risk + Finance + CEO approval.
- **Interface and User Experience**:
    - **Layout**: Dashboard with stage cards (400px wide on desktop, 90% on mobile). Page background: light gray; cards: white. Padding: 32px for cards, 16px for fields.
    - **Colors**: Buttons/links: blue (hover: darker blue); success messages: green; error messages: red; labels: gray; inputs: white with gray border.
    - **Text**: Inter or Roboto font. Headings: 24px, bold; labels/inputs: 16px; messages: 14px (headings: 18px on mobile).
    - **Form Elements**: Full-width inputs/dropdowns, rounded corners. Error/success messages above forms. Submit button at bottom.
    - **Responsiveness**: Stack elements on mobile (below 640px). Follow accessibility standards (ARIA labels, keyboard navigation, high-contrast text).
    - **Branding**: Add Swayatta CRM logo above dashboard, keep design clean and professional.
- **Error Messages and Validation**:
    - **Server-Side**:
        - Missing Sign-off: “Required sign-off missing.”
        - Task Overdue: “Task overdue.”
        - Budget Deviation: “Budget deviation detected.”
        - Change Threshold: “Change exceeds 10% threshold.”
        - Invalid POT-ID/BoQ: “Invalid POT-ID or BoQ.”
        - Unauthorized Access: “Access denied. Required role not found.”
        - Database Failure: “Failed to update deal. Try again.”
- **Industry Standards**:
    - **Security**: Use HTTPS, encrypt sensitive data (e.g., cost models), parameterized queries to prevent SQL injection, JWT for RBAC, sanitize inputs to prevent XSS, log actions securely.
    - **Data Integrity**: Enforce sign-off sequences, prevent deletion of active deals.
    - **Compliance**: GDPR for data handling, PCI-DSS for payment tracking.
    - **Integration**: RESTful APIs for procurement/ERP sync, automated notifications.
- **Non-Functional Requirements**:
    - **Performance**: Load dashboard and process updates in under 2 seconds. Cache stage data, optimize queries.
    - **Scalability**: Handle 10,000 deals with efficient indexing and caching.

    - **Reliability**: Ensure 99.9% uptime, daily backups.
    - **Usability**: Consistent CSS, responsive design.
    - **Compliance**: GDPR/PCI.
- **Scenarios**:
    - Opportunity Qualification: Qualify RFP, trigger DL (N-5).
    - HOTO: Transfer SoW, validate PO (N+1).
    - Change Management: Escalate >10% change to EC with Change Impact Template.
    - Edge Cases: Missing sign-off (blocked), budget overrun (alert), unauthorized access (denied).
- **Technical Instructions**:

Sub-Tasks

- Adding of technical functionalities and generate code
- Deployment
- Fixing of Bugs Manually
- Testing and monitoring