

PROJECT MANAGEMENT APP

YELURI SAHITHI NAVYA

Phase 1: Problem Understanding & Industry Analysis

Goal: Understand the problem, industry practices, and define project objectives clearly.

- Conduct requirement gathering from stakeholders (Project Manager, Clients, Team Leads, Sponsors).
- List down project expectations like task tracking, milestone monitoring, and reporting.
- Analyze stakeholder roles and responsibilities (Admin, Manager, Team, Client).
- Identify pain points in existing manual processes (delays, miscommunication, tracking issues).
- Map business processes: Client request → Task assignment → Progress → Reporting.
- Perform industry research on standard project management practices.
- Benchmark against tools like Jira, Trello to understand best practices.
- Finalize the scope and limitations of the project management app.
- Set measurable success criteria for the solution.

Phase 2: Org Setup & Configuration

Goal: Prepare the Salesforce environment and configure essential settings.

- Sign up for Developer Org for development and testing.
- Configure company profile with time zone, fiscal year, currency settings.
- Define working hours and add official holidays.
- Create users like Project Manager, Team Member, and Client roles.
- Assign profiles to control data visibility and access levels.
- Use Org-Wide Defaults (OWD) for record-level security.
- Set up sharing rules to allow collaboration where needed.
- Create permission sets for additional access without changing profiles.
- Implement login access policies and restrictions for security.

Phase 3: Data Modeling & Relationships

Goal: Build strong data structure with required objects and relationships.

- Identify standard objects (Contacts, Users).
- Create custom objects like Project, Task, Resource Allocation.
- Define key fields: Project Start Date, End Date, Priority, Status.
- Add task-specific fields like Assigned Resource, Deadline, Progress %.
- Create record types for different project types (Agile, Waterfall).
- Design page layouts showing related tasks, resources, and updates.
- Build compact layouts for mobile accessibility.
- Use schema builder to visualize data model and relationships.
- Establish lookup and master-detail relationships as required.

Phase 4: Process Automation (Admin)

Goal: Automate recurring processes and enforce validations.

- Create validation rules (e.g., Task end date must be after start date).
- Use Flow Builder for auto-assigning tasks to resources.
- Implement record-triggered flows for project updates.
- Define approval process for high-priority project tasks.
- Set up email notifications for task assignments and completions.
- Configure custom notifications for in-app alerts.
- Automate reminders for deadlines through scheduled flows.
- Assign follow-up tasks automatically after project phase completion.
- Use conditional logic to update project statuses dynamically.

Phase 5: Apex Programming (Developer)

Goal: Implement advanced logic using Apex where declarative tools are not enough.

- Write Apex classes for task assignment and project scheduling logic.
- Implement triggers to prevent duplicate task assignments.
- Use SOQL queries to fetch real-time project data.
- Apply collections (List, Set, Map) to manage multiple records efficiently.
- Develop Batch Apex to auto-update overdue tasks daily.
- Use Queueable Apex for processing bulk task updates asynchronously.
- Implement Future methods for API callouts to external systems.
- Include robust exception handling for errors in processes.
- Write test classes ensuring at least 75% code coverage.

Phase 6: User Interface Development

Goal: Design a user-friendly UI with Lightning tools and components.

- Build the Project Management App using Lightning App Builder.
- Add tabs for Projects, Tasks, and Resources for easy navigation.
- Customize home page with a dashboard of project KPIs.
- Design record pages with task lists and related project info.
- Create utility bar shortcuts for quick task creation.
- Develop LWC components for searching and filtering tasks.
- Enable navigation service for seamless record access.
- Wire Apex methods to fetch project data into LWC.
- Test responsiveness and accessibility across devices.

Phase 7: Integration & External Access

Goal: Connect Salesforce with external tools and systems.

- Set up Named Credentials for secure API authentication.
- Integrate REST APIs to sync data with tools like Jira or Trello.
- Use platform events for real-time updates across systems.
- Enable Change Data Capture for syncing external systems.
- Use Salesforce Connect to access external databases directly.
- Configure Remote Site Settings for secure callouts.
- Monitor API limits and optimize API usage.
- Set up OAuth for client login through portals.
- Build error-handling mechanisms for integration failures.

Phase 8: Data Management & Deployment

Goal: Manage large sets of data and deploy configurations safely.

- Import demo project and task data using Data Import Wizard.
- Use Data Loader for bulk imports and exports.
- Define duplicate rules to prevent multiple entries of same task.
- Schedule weekly data export for backup.
- Deploy metadata changes using Change Sets.
- Leverage ANT Migration Tool for large-scale deployments.
- Adopt VS Code + Salesforce CLI (SFDX) for version control.
- Test deployments thoroughly in Sandbox before Production.
- Maintain rollback plans for safe deployment.

Phase 9: Reporting, Dashboards & Security Review

Goal: Generate insights and ensure security of project data.

- Create reports for project progress, overdue tasks, and milestones.
- Generate dashboards for managers showing KPIs and resource use.
- Design dynamic dashboards with role-based visibility.
- Define report types linking Projects and Tasks.
- Apply field-level security for sensitive project fields.
- Set login IP ranges and session timeout policies.
- Enable audit trail to monitor admin changes.
- Test data visibility with different profiles and roles.
- Conduct a security review before production release.

Phase 10: Final Presentation & Demo Day

Goal: Deliver the completed solution and demonstrate functionality.

- Prepare a presentation covering problem, solution, and benefits.
- Show live demo of project creation, task assignment, and reports.
- Explain automation flows, triggers, and dashboards.
- Provide user guide and handover documentation.
- Collect feedback from stakeholders during demo.
- Record the demo for future references and portfolio.
- Publish project details on LinkedIn/Portfolio.
- Conduct a Q&A; session to resolve doubts.
- Plan enhancements and future improvements.