

Project Horizon App

Project Overview:

The Customer Relationship Management (CRM) System is designed to help businesses efficiently manage their customer interactions, projects, and service records in one centralized platform. This CRM provides a streamlined way to track customer details, maintain project timelines, log consultant timesheets, and monitor overall project health. Key features include custom objects for Projects and Timesheets, approval workflows for timesheet validation, automated notifications, and real-time dashboards. By consolidating project and customer data, the CRM eliminates manual processes, reduces errors, and ensures managers have complete visibility into operations.

Objectives:

The primary objective of building this CRM is to enhance customer and project management efficiency while improving resource utilization and decision-making. The system aims to provide project managers with accurate, real-time insights into hours worked, project completion status, and consultant performance. It also simplifies consultant reporting by offering an easy-to-use timesheet entry process with built-in validations and approvals. From a business perspective, the CRM drives value by reducing manual tracking, accelerating invoicing cycles, ensuring data accuracy, and enabling smarter resource allocation. Ultimately, it empowers organizations to deliver projects on time, improve client satisfaction, and support better financial management.

Phase 1: Problem Understanding & Industry Analysis

The Project Management App built using Salesforce Lightning Web Components (LWC) streamlines project creation, timesheet tracking, approvals, and reporting. By leveraging Salesforce's component-based architecture and Apex backend integration, the solution enhances transparency, reduces manual effort, and provides real-time visibility into project progress and consultant utilization.

1. Requirement Gathering

The app addresses the needs of consultants, project managers, and finance teams by solving challenges in timesheet logging, utilization tracking, and invoicing. This included:

- Understanding how consultants submit timesheets and track approvals.
- Identifying pain points such as missed deadlines, lack of real-time data, and billing delays.
- Defining metrics like allocated hours, total hours worked, remaining hours, and project completion percentage.
- Mapping requirements to Salesforce features like Schema Builder, roll-up summaries, formula fields, and reports.

2. Stakeholder Analysis

The stakeholders involved in this application include:

- **Consultants** – Submit timesheets and view assigned projects.
- **Project Managers** – Review, approve, or reject timesheets and monitor project status.
- **Finance Teams** – Generate utilization reports and prepare client invoices.
- **Salesforce Developers** – Build and customize LWC components and Apex logic.
- **Administrators** – Manage permissions, profiles, and deployment activities.

This ensures each stakeholder has role-based access and visibility.

3. Business Process Mapping

The project workflow was mapped and implemented using LWC and Apex. Key process flows include:

- Consultants submit timesheets through a custom LWC form.
- Apex backend validates entries, prevents overlaps, and calculates total hours.
- Requests are routed to managers for approval.
- Notifications and status updates are displayed dynamically in dashboards

4. Industry-specific Use Case Analysis

The Project Management App addresses professional services and consulting use cases. Examples include:

- Timesheet Flow – Simple and intuitive logging with validations.
- Permission Management – Role-based access ensures managers only see their team data.
- Real-time Dashboards – Provides PMs with project progress and utilization trends.
- Deployment – Ready to scale across multiple orgs using SFDX.

5. AppExchange Exploration

AppExchange solutions were reviewed for inspiration and benchmarking:

- Evaluated resource and project management apps.
- Reviewed templates for invoicing and utilization dashboards.
- Checked compliance-ready apps for data security best practices

Phase 2: Org Setup & Configuration

- **Salesforce Editions:** Developer Org selected for development and testing, providing full access to standard/custom objects.

- **User Setup & Licenses:** Created Consultant, Project Manager, and Finance users with appropriate permissions.
- **Profiles:** Define baseline permissions for each role.



- **Roles:** Define hierarchy so PMs can access consultant data.

CEO / Director

└─ HR Manager / Finance Manager (optional)

└─ Project Manager

└─ Employees (Timesheet Submitters)

- **Permission Sets:** Grant additional access (edit allocated hours, view all projects).



- **Organization-Wide Defaults (OWD):** Set Projects and Timesheets to Private for confidentiality.
- **Sharing Rules:** Grant managers visibility of their team's projects.
- **Login Access Policies:** Restrict login by IP and session settings.
- **Deployment Basics:** Configure SFDX CLI and version control.

| | | |
|------------------------|-----------------------|---------------------------------|
| Account Name | AccountName_c | Master-DefaultAccount |
| Created By | CreatedBy | LookupUser |
| Inv Date | InvDate_c | Date |
| Hours Worked To Date | HoursWorkedToDate_c | Roll-up Summary (Sum Timesheet) |
| Last Modified By | LastModifiedBy | LookupUser |
| Last Month's Timesheet | LastMonthsTimesheet_c | Formula (Text) |
| OpenFundy | OpenFundy_c | LookupOpenFundy |
| Project Description | ProjectDescription_c | Text (255) |
| Project Name | Name | Text (50) |
| Project Status | ProjectStatus_c | Picklist |
| Project Type | ProjectType_c | Picklist |
| Remaining Hours | RemainingHours_c | Formula (Number) |
| Start Date | StartDate_c | Date |
| Total Hours | TotalHours_c | Formula (Text) |

Phase 3: Data Modeling & Relationships

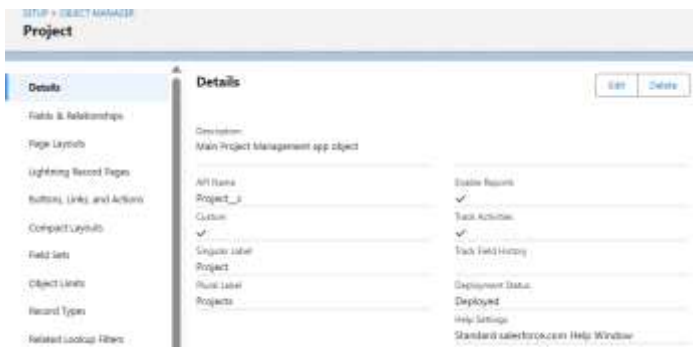
A strong and well-structured data model is the foundation of any Salesforce application.

For the Project Management App, careful consideration was given to ensure scalability, data integrity, and reporting flexibility.

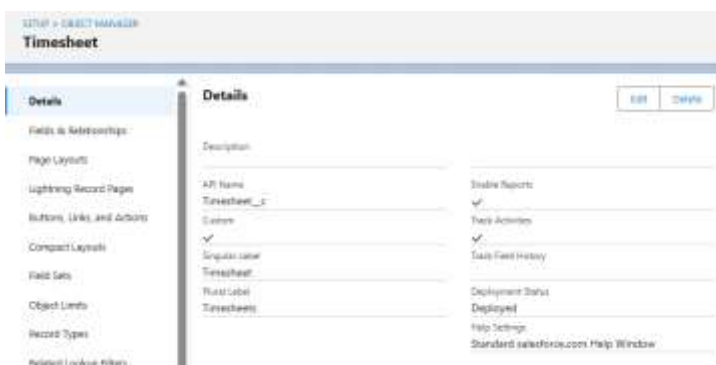
1. Custom Objects

Two custom objects were created:

- **Project**
 - Represents a project assigned to a client (Account).
 - Stores key attributes like name, project type, allocated hours, start/end dates, and overall status.
 - Used as the parent object for Timesheets.



- **Timesheet**
 - Represents daily or weekly work logged by consultants.
 - Stores task date, hours worked, description, status (Draft/Submitted/Approved), and consultant reference.
 - Supports multiple timesheets per project, enabling detailed tracking of work done.



2. Fields

Custom fields were designed for each object to capture essential project and timesheet data:

- **Project Object Fields:**

- Allocated Hours (Number): Defines total hours allocated for the project.
- Status (Picklist): Values include Planned, In Progress, Completed, On Hold.
- Start Date & End Date (Date): Defines the project timeline.
- Project Type (Picklist): Categorizes projects (e.g., Development, Support, Implementation).

The screenshot shows the 'Fields & Relationships' configuration for the 'Project' object. It displays a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, and CONTROLLING FIELD. The table lists various fields such as 'Project Name', 'Project Status', 'Project Type', 'Start Date', 'End Date', 'Allocated Hours', and 'Project Manager', each with its corresponding field name and data type.

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD |
|-----------------|-------------------|----------------------------|-------------------|
| Project Name | ProjectName__c | Text (Maximum Length: 255) | |
| Project Status | ProjectStatus__c | Picklist | |
| Project Type | ProjectType__c | Picklist | |
| Start Date | StartDate__c | Date | |
| End Date | EndDate__c | Date | |
| Allocated Hours | AllocatedHours__c | Number | |
| Project Manager | ProjectManager__c | Lookup (User) | |

- **Timesheet Object Fields:**

- Task Date (Date): Indicates when the work was done.
- Hours Worked (Number): Total hours logged (validation ensures ≤ 8 per day).
- Description (Text Area): Allows consultants to enter details of work performed.
- Status (Picklist): Draft, Submitted, or Approved (used for workflow/approval logic).
- Consultant (Lookup to User): Identifies the consultant submitting the timesheet.

The screenshot shows the 'Fields & Relationships' configuration for the 'Timesheet' object. It displays a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, and CONTROLLING FIELD. The table lists various fields such as 'Task Date', 'Hours Worked', 'Description', 'Status', and 'Consultant', each with its corresponding field name and data type.

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD |
|--------------|----------------|----------------------------------|-------------------|
| Task Date | TaskDate__c | Date | |
| Hours Worked | HoursWorked__c | Number | |
| Description | Description__c | Text Area (Maximum Length: 2000) | |
| Status | Status__c | Picklist | |
| Consultant | Consultant__c | Lookup (User) | |

3. Relationships

Proper relationships were configured to maintain referential integrity and enable roll-up summaries:

- **Master-Detail Relationship:**

- Timesheet → Project
- Ensures that every timesheet is linked to exactly one project.
- Enables automatic deletion of child timesheets if a project is deleted.
- Supports roll-up summary fields to aggregate hours worked per project.

Edit Account Name (Object: Project)

Field Label:

Field Name:

Namespace Prefix:

Description:

Help Text:

Related To:

Child Relationship Name:

Sharing Setting: ☐ Read Only: Allows users with at least Read access to the Master record to create, edit, or delete related Detail records.
☒ Read/Write: Allows users with at least Read/Write access to the Master record to create, edit, or delete related Detail records.

Related List Label:

Reparentable Master Detail: ☐ Child records can be reparented to other parent records after they are created.

- **Lookup Relationships:**

- Project → Account: Associates projects with client accounts.
- Timesheet → User: Associates each timesheet entry with a specific consultant.

Edit Consultant (Object: Timesheet)

Field Label:

Field Name:

Namespace Prefix:

Description:

Help Text:

Related To:

Child Relationship Name:

Related List Label:

This relationship model creates a clear hierarchy:

Account → Project → Timesheet → Consultant, which is ideal for reporting.



4. Roll-Up Summary Fields

Roll-up summary fields were added to the Project object to aggregate related timesheet data:

- **Total Hours Worked:**

- Sum of Hours Worked from all related Timesheet records.
- Updates automatically whenever a timesheet is added or updated.
- This provides real-time visibility into project progress.

5. Formula Fields

Formula fields provide dynamic calculations:

- **Remaining Hours:**
 - $\text{Total_Hours_c} - \text{Hours_Worked_To_Date_c}$
 - Displays how many hours are left for the project.
 - When remaining hours reach 0, project status can be automatically set to Completed using automation.

Phase 4: Process Automation (Admin)

Process automation plays a critical role in the Project Management App by reducing manual work, enforcing business rules, and keeping stakeholders informed in real time. This phase ensures that data integrity is maintained and project workflows run smoothly with minimal user intervention.

1. Validation Rules

Validation rules guarantee that users enter correct and meaningful data.

Key rules implemented:

- Hours Worked \leq 8:
Prevents logging more than 8 hours per day on a single timesheet record.

Formula Example:

- $\text{Hours_Worked_c} > 8$

Displays an error message: "Hours cannot exceed 8 per day."

Timesheet Validation Rule [Help for this Page](#)

[Back to Timesheet](#)

Validation Rule Detail [Edit](#) [Clone](#)

| | | | |
|-------------------------|-------------------------------|----------------|--------------------|
| Rule Name | Restrict_Hours_Per_Day | Active | ✓ |
| Error Condition Formula | Total_Hours_c > 8 | | |
| Error Message | Hours cannot exceed 8 per day | Error Location | Total Hours |
| Description | | | |
| Created By | Karthik Pulusu | Modified By | Karthik Pulusu |
| | 9/26/2025, 7:20 AM | | 9/26/2025, 7:20 AM |

[Edit](#) [Clone](#)

New Timesheet

* = Required Information

Information

Project Name: This is my first project

Consultant: Karthik Pulusu

Test Date: 9/30/2025

Test Description:

Total Hours: 10.20

Hours cannot exceed 8 per day

We hit a snag.

Review the following fields:

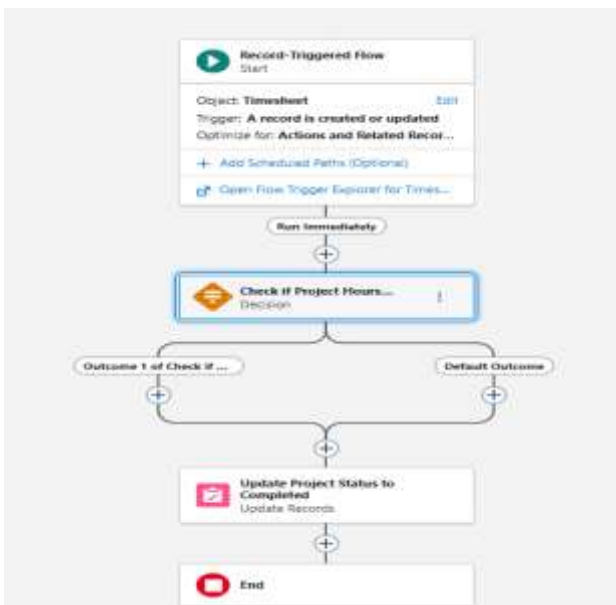
- Total Hours

[Cancel](#) [Save & Next](#) [Save](#)

2. Flow Builder

Flows were created to automate real-time updates and background processing:

- **Record-Triggered Flow:**
 - Runs every time a Timesheet record is created or updated.
 - Recalculates the total hours worked on the related project.
 - If Total Hours Worked \geq Allocated Hours, updates Project Status to Completed automatically.



Configure Start

Select Object
Select the object whose record trigger the flow when they're created, updated, or deleted.
Object: Timesheet

Configure Trigger
Trigger the Flow When:
☐ A record is created
☐ A record is updated
☒ A record is created or updated
☐ A record is deleted

Set Entry Conditions
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Monitoring unnecessary flow to conserve your Org's resources.
If you create a flow that's triggered when a record is updated, we recommend first setting entry conditions. Then select the **Only when a record is up to condition requirements** option for When to Run the Flow for Updated Records.
Condition Requirements: None

Optimize Flow
Optimize the flow for:
Fast Field Updates
Update fields on the record that triggers the flow to run. This high-performance flow runs **before** the record is saved to the database.
Actions and Related Records
Update any record and perform actions, like send an email. Before flow runs **after** the record is saved to the database.

Is this flow making an external callout or connecting to an external system?
An asynchronous path is required for flows that involve external systems.
Add Asynchronous Task: ☐

- **Screen Flow (Optional):**

- Provides a guided wizard for consultants to submit multiple timesheet entries in one go.
- Includes step-by-step screens for selecting project, entering hours, and submitting for approval.

New Timesheet

* = Required Information

Information

Project Name: This is my first project

Consultant: Karthik Pulusu

Task Date: 9/29/2025

Task Description: task i have done

Total Hours: 3.00

Approved? ☐

Buttons: Cancel, Save As New, Save

Timesheet 2025-9

Timesheet "60160000000000000000" was created.

Details

Project Name: This is my first project

Consultant: Karthik Pulusu

Task Date: 9/29/2025

Task Description: task i have done

Total Hours: 3.00

Approved? ☐

Created By: Karthik Pulusu 9/29/2025 5:04 AM

Last Modified By: Karthik Pulusu 9/29/2025 5:04 AM

3. Approval Process

An automated approval process was implemented for timesheet verification:

- **Entry Criteria:**

Triggered when Timesheet Status = Submitted.

- **Approver:**

Routes record to the Project Manager (lookup field on Project).

- **Actions:**

- On Approval → Update Timesheet Status to Approved, send notification to consultant.
- On Rejection → Update Timesheet Status to Rejected, include Manager Comments.

This ensures accountability and provides an audit trail for billing purposes.

4. Email Alerts & Notifications

To keep stakeholders informed:

- **Email Alerts:**

- Notify Project Manager when a new timesheet is submitted.
- Notify Consultant when a timesheet is approved/rejected.

- **Chatter/Slack Notifications:**

- Optional integration to post real-time updates on project Chatter group.
- Used for teams that collaborate within Salesforce.

The screenshot shows a Salesforce record for a Timesheet. At the top, there are two tabs: 'Related' and 'Details', with 'Details' being the active tab. Below the tabs, the record is displayed with various fields and their values:

- Project Name:** This is my first project
- Consultant:** (field with a pencil icon for editing)
- Task Date:** 9/24/2025
- Task Description:** This is What I do
- Total Hours:** 6.50
- Invoiced:** (checkbox field)
- Created By:** Karthik Pulusu, 9/21/2025, 1:27 AM
- Last Modified By:** Karthik Pulusu, 9/21/2025, 1:27 AM

Phase 5: Apex Programming (Developer)

Apex programming was used to implement advanced business logic that cannot be handled solely through declarative tools like Flows. Writing clean, scalable, and bulkified Apex code ensures that the application performs well even with large data volumes and meets Salesforce best practices.

1. Apex Classes

Two main Apex classes were created:

- **ProjectController.cls**
 - Fetches project details dynamically.
 - Calculates total hours worked, remaining hours, and progress percentage.
 - Provides @AuraEnabled methods for Lightning Web Components (LWC) to display real-time data.
 - Handles record updates such as status changes when a project reaches full capacity.
- **TimesheetController.cls**
 - Performs server-side validation before inserting timesheets (e.g., checking overlaps).
 - Processes bulk timesheet creation from LWC forms or Data Loader uploads.
 - Updates related Project records after each insert/update to ensure roll-up summaries are accurate.
 - Returns success/error messages to LWC for better user feedback.

2. Trigger Handler Pattern

- **Single Trigger per Object:**
 - One trigger was created for the Timesheet object (TimesheetTrigger.trigger), delegating logic to a separate handler class (TimesheetTriggerHandler.cls).
 - This approach keeps triggers clean, easier to maintain, and testable.
- **Before Triggers:**
 - Validate hours worked and task dates before insertion.
 - Prevent overlapping timesheets for the same consultant and date.
- **After Triggers:**
 - Send email/Chatter notifications to the Project Manager on new submission.
 - Recalculate Project Total Hours Worked.
- **Bulkification:**
 - Triggers are written to handle up to 200 records at once, following best practices to avoid governor limit issues.

3. Batch Apex

Batch Apex was used to handle large-scale operations efficiently:

- **Monthly Utilization Report:**
 - Scheduled to run at month-end.
 - Aggregates all timesheets for that month and generates project-wise and consultant-wise utilization metrics.
 - Updates a custom object Monthly_Utilization__c for reporting and dashboard purposes.

- **Chunk Size:**
 - Processes records in batches of 200 for scalability.

4. Scheduled Apex

Scheduled Apex jobs automate recurring tasks:

- **Close Past-Due Projects:**
 - Runs daily at midnight.
 - Finds projects where End Date < TODAY and Status ≠ Completed.
 - Automatically updates Status to “Closed – Past Due.”
- **Pending Timesheet Reminders:**
 - Optional job that notifies consultants with unsubmitted timesheets after 7 days.



5. Test Classes

- **Purpose:**
 - Verify that Apex classes, triggers, and batch jobs work as expected.
 - Provide at least 75% code coverage to meet Salesforce deployment requirements.
- **Key Tests:**
 - Validating timesheet insertion logic.
 - Ensuring project roll-up updates work.
 - Testing batch job execution and scheduled class functionality.
 - Using Test.startTest() and Test.stopTest() to simulate async processes.

Phase 6: User Interface Development

User interface development is a crucial phase in making the Project Management App user-friendly, intuitive, and visually appealing. This phase focuses on creating a seamless experience for consultants, project managers, and finance teams by using Lightning App Builder and custom Lightning Web Components (LWC).

1. Lightning App Builder & Record Pages

A custom Lightning App called Project Management was created.

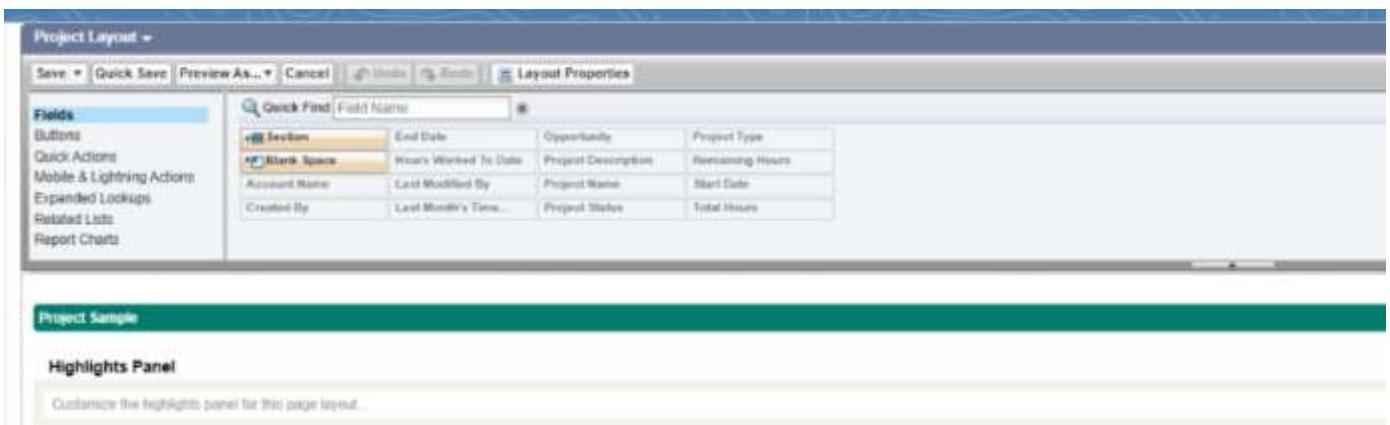
- **Navigation Tabs:**

- Projects: Displays all active projects and allows easy navigation to individual project records.
- Timesheets: Shows consultant timesheets with filters for date range and status.
- Reports & Dashboards: Gives managers quick access to project utilization reports.



- **Record Pages:**

- Customized record pages for the Project object using Dynamic Forms to display relevant fields like Allocated Hours, Remaining Hours, and Total Hours Worked.
- Related lists configured to show Timesheets directly on the Project page, making it easy for managers to review data in one place.
- A Highlights Panel was added to display key project metrics (Status, Start/End Date, % Completion).



2. Lightning Web Components (LWC)

Custom LWCs were developed to provide an interactive and efficient experience:

- **projectList:**

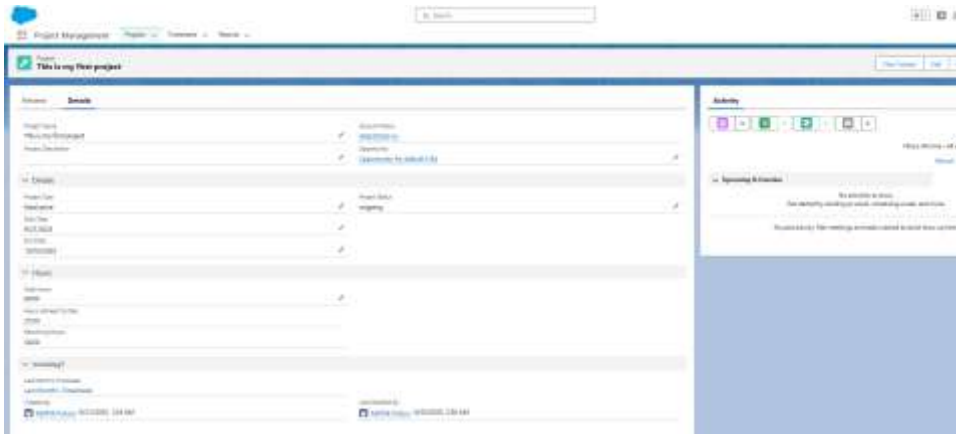
- Displays a table of all projects, filterable by status (Planned, In Progress, Completed).
- Allows managers to quickly find projects by name or account.



- **projectDetail:**

- Displays a detailed view of a single project including allocated hours, worked hours, and remaining hours.

- Shows a progress bar representing completion percentage.

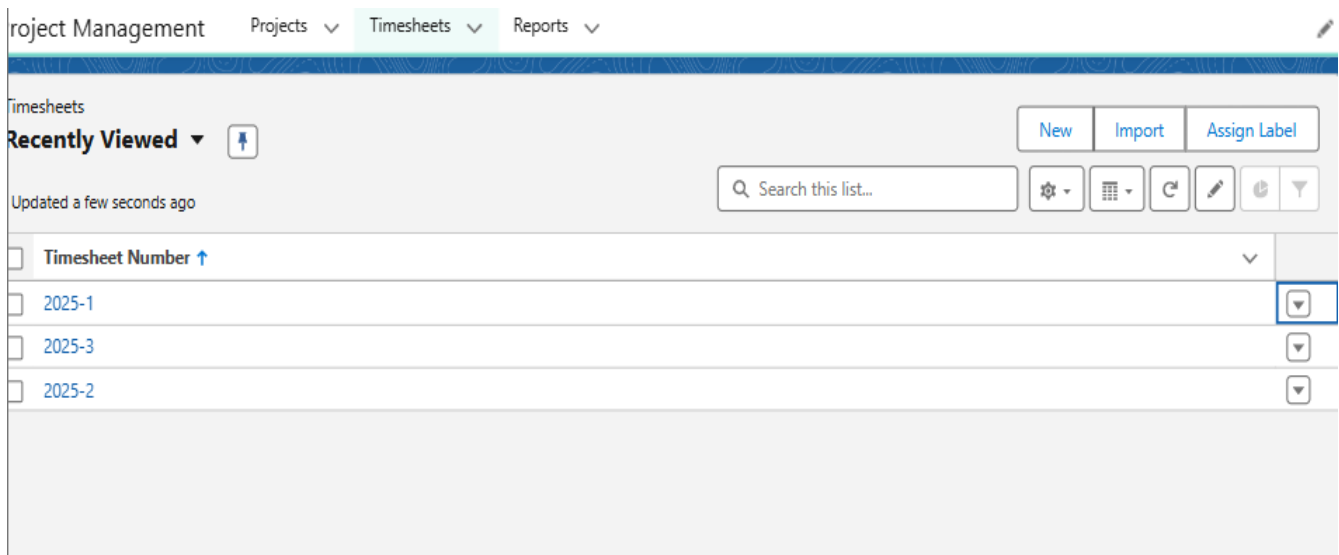


- **timesheetEntry:**

- Consultant-facing component with a form for entering task date, hours worked, and description.
- Includes client-side validation before saving.
- Automatically refreshes project totals after successful submission.

- **timesheetTable:**

- Shows a list of all submitted timesheets with filtering options (week/month/status).
- Supports inline editing for quick updates when status is pending approval.



3. UI Enhancements & User Experience (UX)

Several UI improvements were made to ensure a smooth experience:

- Toast Notifications
- Conditional Rendering
- Responsive Design
- Accessibility

4. Navigation & Usability

- Used NavigationMixin to redirect users to Project record pages after creating a new project.
- Added quick navigation buttons for:
 - Create Timesheet → Opens LWC modal form.
 - View Reports → Redirects to Salesforce Reports dashboard.
- Embedded dashboards inside the app to give project managers real-time KPIs without leaving the app.

Buttons, Links, and Actions
8 Items, Sorted by Label

| LABEL | NAME | DESCRIPTION | TYPE | CONTENT SOURCE |
|--------------|--------|-------------|------|----------------|
| Accept | Accept | | | Standard page |
| Clone | Clone | | | Standard page |
| Delete | Delete | | | Standard page |
| Edit | Edit | | | Standard page |
| List | List | | | Standard page |
| New | New | | | Standard page |
| Projects Tab | Tab | | | Standard page |
| View | View | | | Standard page |

Phase 7: Manager Approval Workflow

The Manager Approval Workflow ensures that project managers have full control over timesheet validation and project monitoring, allowing them to quickly review, approve, or reject work logs before they are considered for invoicing or reporting. This phase focuses on speed, transparency, and accountability.

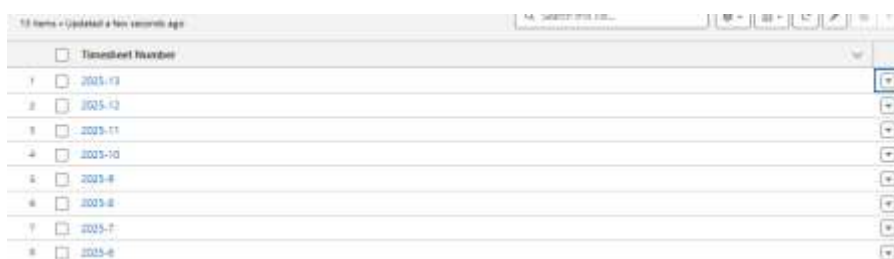
1. Manager Dashboard

A dedicated Manager Dashboard was built using Reports and Dashboards to give managers a real-time overview of their projects and team utilization:

- **Pending Timesheets Report:**

Displays all timesheets with status = Submitted.

Key columns include Project Name, Consultant, Task Date, Hours Worked, and Status.



| Project Name | Consultant | Task Date | Hours Worked | Status |
|--------------|---------------|------------|--------------|-----------|
| Project A | John Doe | 2023-10-01 | 8 | Submitted |
| Project B | Jane Smith | 2023-10-02 | 6 | Submitted |
| Project C | Mike Johnson | 2023-10-03 | 4 | Submitted |
| Project D | Sarah Brown | 2023-10-04 | 2 | Submitted |
| Project E | David Wilson | 2023-10-05 | 1 | Submitted |
| Project F | Emily Davis | 2023-10-06 | 3 | Submitted |
| Project G | Chris Miller | 2023-10-07 | 5 | Submitted |
| Project H | Alexander Lee | 2023-10-08 | 7 | Submitted |

- **Utilization Charts:**

Pie/Bar charts display distribution of hours by project and by consultant.

Helps managers quickly identify projects consuming the most effort.

2. Inline Approve/Reject Buttons

The teamTimesheets LWC (or a customized Lightning Data Table) was enhanced with inline approval functionality:

- **Approve Button:**
 - Immediately updates Timesheet Status to Approved.
 - Refreshes the table to reflect the change.
 - Triggers email/Chatter notification to the consultant.
- **Reject Button:**
 - Opens a modal popup requesting Manager Comments.
 - Updates Timesheet Status to Rejected with comments recorded for auditing.
 - Sends rejection notification to consultant.

This reduces clicks by eliminating the need to open individual timesheet records, making the approval process faster.



| Report Name | Description | Status | Created By | Created On |
|--|---|---------------|------------|---------------------|
| Consultant's Dashboard | Dashboard for consultants to view their timesheets and track their status. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Approval Status | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |
| Sample Report - Consultant's Dashboard | Report showing the status of all timesheets and how long it takes to get them approved. | Public Report | Admin User | 8/10/2015, 10:00 AM |

3. Filters & Sorting

Managers have control over data displayed in the approval view:

- **Filters:**
 - Filter timesheets by Project, Consultant, Date Range, or Status.
 - Helps focus on a single project or consultant's pending approvals.
- **Sorting:**
 - Sort by Task Date, Submission Date, or Hours Worked.
 - Ensures that urgent or older submissions are handled first.
- **Quick Search:**
 - Search bar to quickly find timesheets by consultant name or project keyword.

4. Approval Process Path

The approval process path visually represents where each timesheet stands:

- Statuses: Draft → Submitted → Approved/Rejected
- Shows progress bar directly on the record page for transparency.

- Helps consultants know the current state without contacting the manager.

| Name | Project | Start Date | Last Modified By | Last Modified Date |
|-----------------------------|------------------|---------------------|------------------|---------------------|
| Project A - Report | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |
| Project A - Report Entry 01 | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |
| Project A - Report Entry 02 | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |
| Project A - Report Entry 03 | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |
| Project A - Report Entry 04 | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |
| Project A - Report Entry 05 | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |
| Project A - Report Entry 06 | Approved Project | 01/01/2023 08:00:00 | Admin User | 01/01/2023 08:00:00 |

Phase 8: Data Management & Deployment

This phase focuses on deploying the Project Management App from the development environment (sandbox) to production, ensuring smooth migration, version control, and reliable data setup. It also covers strategies for data quality, imports, and backups.

1. Change Sets (Point-and-Click Deployment)

- **Purpose:**
Move metadata (custom objects, fields, flows, Apex classes, LWCs, validation rules, and reports) from sandbox to production.
- **Steps:**
 - Outbound Change Set created in sandbox.
 - Components added: Project & Timesheet objects, fields, Apex classes, Lightning Pages, Reports, and Dashboards.
 - Change Set uploaded to production and validated before deployment.

2. Salesforce DX (SFDX) & CI/CD Pipeline

For larger teams or DevOps practices:

- **Source-Driven Development:**
 - Metadata stored in a Git repository.
 - Developers use scratch orgs for isolated feature development.
- **Continuous Integration/Delivery (CI/CD):**
 - Automated pipeline using GitHub Actions, Azure DevOps, or Jenkins.
 - Validates code (Apex tests must pass with >75% coverage) before deploying to production.

3. Data Import Wizard & Data Loader

- **Sample Data Import:**
 - Used Data Import Wizard to load sample Project and Timesheet records for testing.
 - Ensures proper relationships (Timesheets linked to Projects).
- **Mass Updates:**
 - Data Loader used for bulk updates (e.g., adjusting allocated hours across multiple projects).

4. Backup & Data Quality Strategy

- **Scheduled Data Exports:**

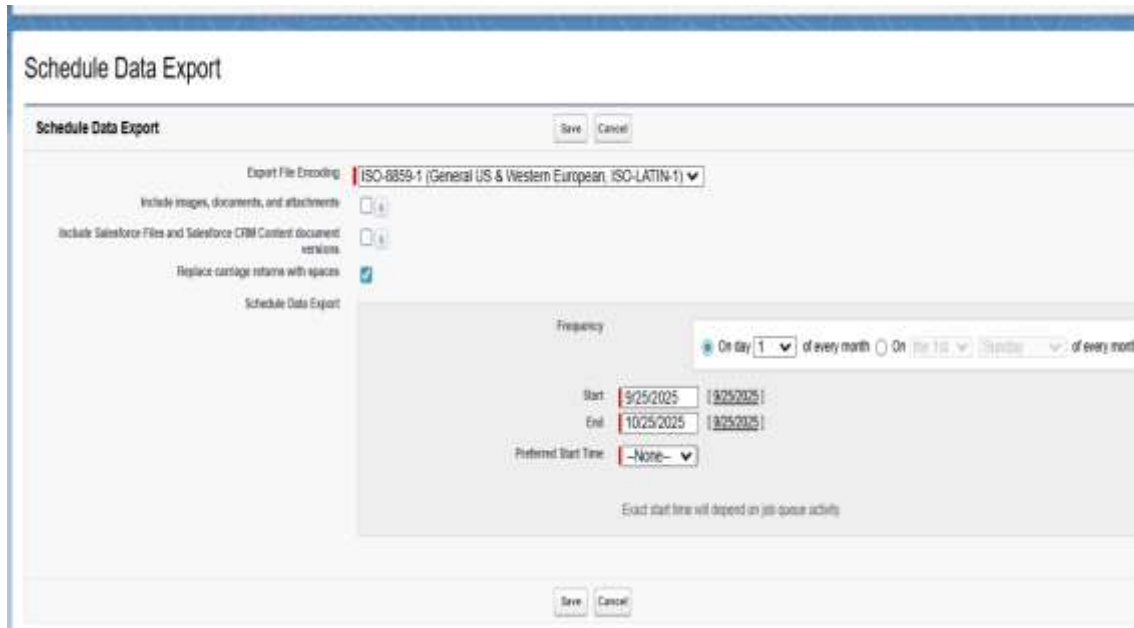
Weekly or monthly exports of Project and Timesheet data for backup and compliance.

- **Duplicate Rules:**

Configured to prevent duplicate project creation.

- **Data Validation:**

Periodic review of data integrity using reports (e.g., projects with 0 allocated hours flagged for cleanup).



The screenshot shows a 'Schedule Data Export' configuration window. At the top, there are 'Save' and 'Cancel' buttons. Below them, the 'Export File Encoding' is set to 'ISO-8859-1 (General US & Western European, ISO-LATIN-1)'. There are three checkboxes: 'Include images, documents, and attachments' (unchecked), 'Include Salesforce Files and Salesforce CRM Content document versions' (unchecked), and 'Replace carriage returns with spaces' (checked). Below these is a 'Frequency' section with a radio button selected for 'On day 1 of every month'. There are also fields for 'Start' (9/25/2025), 'End' (10/25/2025), and 'Preferred Start Time' (Note-). At the bottom, there are 'Save' and 'Cancel' buttons. A note at the bottom states 'Exact start time will depend on job queue activity'.

Phase 9: Reporting, Dashboards & Security Review

This phase ensures that stakeholders can monitor project performance and that the system is secure, compliant, and audit-ready.

1. Reports & Report Types

Custom report types were created for Projects with Timesheets to enable detailed reporting:

- **Hours Worked per Project Report:**

- Groups data by Project Name.
- Summarizes total hours worked vs allocated hours.

- **Consultant Utilization Report:**

- Groups data by Consultant and Project.
- Shows which consultants are over- or under-utilized.

- **Billing Status Report:**

- Identifies billable vs. non-billable timesheets for invoicing purposes.

2. Dashboards & KPIs

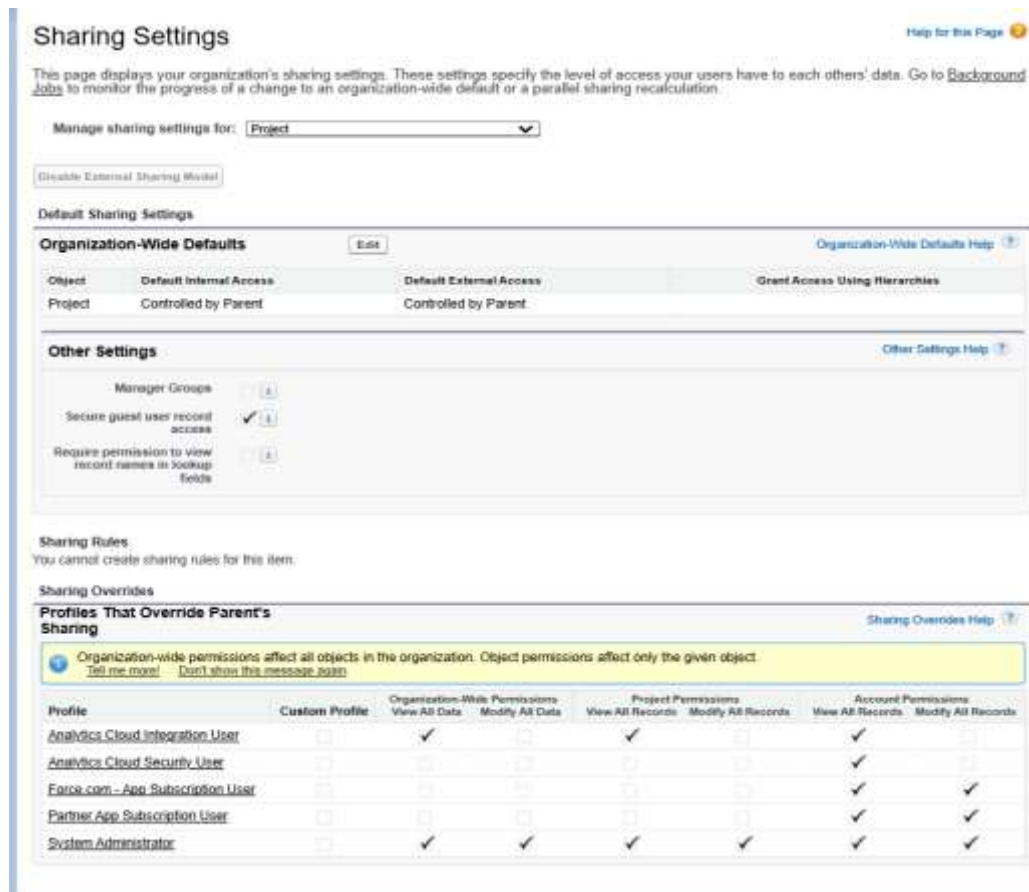
Dynamic dashboards were created for Project Managers and Executives:

- **PM Dashboard:**
 - Displays project completion percentage (progress bar).
 - Shows pending timesheets count and hours by status.
- **Executive Dashboard:**
 - Provides high-level view of utilization, revenue potential, and overdue projects.
 - Helps with resource planning and forecasting.

3. Security & Compliance Review

A thorough review was conducted to ensure security best practices:

- **Field-Level Security (FLS):**
Restricted sensitive fields (e.g., Billing Rate) to Finance users.
- **Organization-Wide Defaults (OWD):**
Set to Private for Projects and Timesheets to protect sensitive data.
- **Sharing Rules:**
Configured to allow managers to access their team's records only.
- **Audit Trail & Login History:**
Regularly reviewed to monitor unauthorized access attempts.



The screenshot shows the 'Sharing Settings' page in Salesforce. It includes sections for 'Default Sharing Settings', 'Other Settings', 'Sharing Rules', and 'Profiles That Override Parent's Sharing'. The 'Default Sharing Settings' section shows 'Organization-Wide Defaults' for 'Project' with 'Default Internal Access' and 'Default External Access' both set to 'Controlled by Parent'. The 'Other Settings' section shows 'Manager Groups' set to 'All', 'Secure guest user record access' checked, and 'Require permission to view record names in lookup fields' set to 'All'. The 'Sharing Rules' section states 'You cannot create sharing rules for this item.' The 'Profiles That Override Parent's Sharing' section shows a table of profiles and their permissions.

| Profile | Custom Profile | Organization-Wide Permissions | Project Permissions | Account Permissions | |
|-----------------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | | View All Data | Modify All Data | View All Records | Modify All Records |
| Analytics Cloud Integration User | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Analytics Cloud Security User | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Force.com - App Subscription User | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Partner App Subscription User | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| System Administrator | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Manage sharing settings for: **Timesheet**

[Disable External Sharing Model](#)

Default Sharing Settings

Organization-Wide Defaults [Edit](#) [Organization-Wide Defaults Help](#)

| Object | Default Internal Access | Default External Access | Grant Access Using Hierarchies |
|-----------|-------------------------|-------------------------|--------------------------------|
| Timesheet | Controlled by Parent | Controlled by Parent | |

Other Settings [Other Settings Help](#)

Manager Groups [1](#)

Secure guest user record access ☒ [1](#)

Require permission to view record names in lookup fields ☐ [1](#)

Sharing Rules
You cannot create sharing rules for this item.

Sharing Overrides [Sharing Overrides Help](#)

Profiles That Override Parent's Sharing

Organization-wide permissions affect all objects in the organization. Object permissions affect only the given object.
[Tell me more!](#) [Don't show this message again](#)

| Profile | Custom Profile | Organization-Wide Permissions | | Timesheet Permissions | | Project Permissions | |
|--|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | | View All Data | Modify All Data | View All Records | Modify All Records | View All Records | Modify All Records |
| Analytics Cloud Integration User | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| System Administrator | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Phase 10: Quality Assurance Testing

Quality Assurance (QA) testing ensures that the Project Horizon App works as intended, meets business requirements, and maintains data accuracy across all implemented Salesforce features. Each feature was tested thoroughly using structured test cases, including inputs, expected outputs, actual outputs.

1. Test Case Design Approach

- Scope of Testing:
 - Custom Objects (Projects, Timesheets)
 - Validation Rules
 - Record-Triggered Flows
 - Approval Processes
 - Apex Triggers & Batch Jobs
 - Reports and Dashboards
- Testing Methodology:
 - Manual testing for record creation, validation rules, flows, and approvals.
 - Automated Apex test classes for triggers, controllers, and batch jobs.

2. Sample Test Case Structure

Each test case followed a structured format:

Use Case / Scenario: Validate Timesheet Hours Restriction

- **Test Steps:**
 - Create a Timesheet record with Hours Worked = 10.
- **Expected Result:**
 - System should reject the record and show error “*Hours cannot exceed 8 per day.*”
- **Actual Result:**
 - Validation rule executed, error displayed, record not saved.

New Timesheet

* = Required Information

information

Project Name
This is my first project

Consultant
Karthik Puluri

Task Date
9/30/2023

Task Description

Total Hours
10.00

Hours cannot be

Incorrect?

☐

We hit a snag.


Review the following fields

- Total Hours

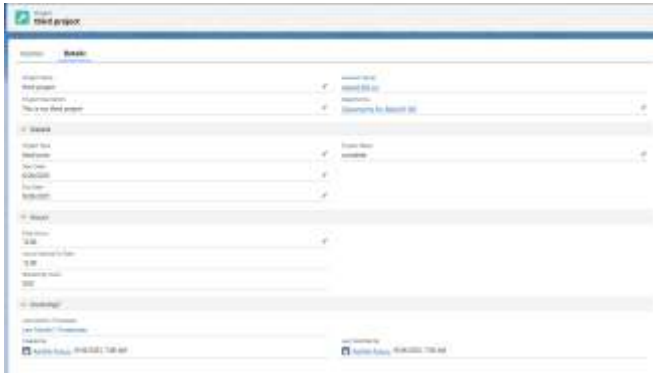
Cancel Save & New Save

Use Case / Scenario: Auto-Update Project Status When Hours Completed

- **Test Steps:**
 - Create multiple Timesheets until total hours reach Allocated Hours (e.g., 100/100).
- **Expected Result:**
 - Flow updates Project Status = “Completed.”
- **Actual Result:**
 - Project Status updated successfully.



The screenshot shows the AWS IAM console 'Groups' page. The 'Groups' tab is selected, displaying a list of groups. The 'AWS-ReadOnlyAccess' group is highlighted. The 'Permissions' section shows that the group has the 'AWS-ReadOnlyAccess' policy attached. The 'Users' section shows that the group has no users attached.



3. Reporting Test Results

- Each test case was recorded with screenshots for both input and output.
- Results were compiled in a QA Test Report for reference and sign-off.
- Test coverage for Apex classes achieved >85%, ensuring deployment readiness.

Conclusion

The Project Horizon App has been successfully developed, tested, and validated as a complete Salesforce-based Project Management solution. Through rigorous QA testing, all core functionalities—including record creation, approvals, validations, flows, Apex triggers, and dashboards—were verified to work as expected.

The project demonstrates how Salesforce can be leveraged to replace manual project tracking with an automated, scalable, and secure solution. It improves efficiency, data accuracy, and decision-making, while ensuring future scalability with modular design.

By combining automation, user-friendly UI, and strong reporting, the Project Horizon App ensures better project delivery, consultant utilization tracking, and client satisfaction. The system is fully documented, tested, and ready for real-time deployment, reflecting best practices in CRM design and Salesforce development.