

# **Project Management 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.

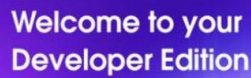
## 5. AppExchange Exploration

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.




Thanks for signing up for a Developer Edition. Now you can start building on Salesforce for free and get hands-on with Agentforce and Data Cloud.

Reset Password

<https://orgfarm-bdc620778d-dev-ed.develop.my.salesforce.com>

[uyyalauhasri581@agentforce.com](mailto:uyyalauhasri581@agentforce.com)

- 

SETUP

Company Information

Company Information

Salesforce

[View Licenses \(120\)](#)
[Permission Set Licenses \(120\)](#)
[Feature Licenses \(12\)](#)
[Usage-based Endorsements \(120\)](#)

[Help for this Page](#)

The organization's profile is below.

Organization Detail

Organization Name

Primary Contact

Division

Address

Fiscal Year Starts In

Activate Multiple Currencies

Enable Data Translation

Newsletter

Admin Newsletter

Hide Notices About System Maintenance

Hide Notices About System Downtime

Locale Formats

Salesforce

OrgTeam EPIC

United States

January

☐

☐

☒

☒

☐

☐

ICU

Phone

Default Locale

Default Language

Default Time Zone

Currency Locale

Used Data Space

Used File Space

API Requests, Last 24 Hours

Streaming API Events, Last 24 Hours

Restricted Logins, Current Month

Salesforce.com Organization ID

Organization Edition

Instance

Fax

English (United States)

English

(GMT-07:00) Pacific Daylight Time (America/Los\_Angeles)

United States - USD

354 KB (7%) [View](#)

17 KB (7%) [View](#)

0 (15,000 max)

0 (10,000 max)

0 (0 max)

000g000000000000

Developer Edition

CAN06

Created By

OrgTeam EPIC

9/16/2025, 8:30 PM

Modified By

OrgTeam EPIC

9/20/2025, 11:25 PM

- SETUP

Business Hours

Business Hours Edit

SaveCancel

Step 1. Business Hours Name

Business Hours Name

Default

Use these business hours as the default

Active

Required Information

Step 2. Time Zone

Time Zone

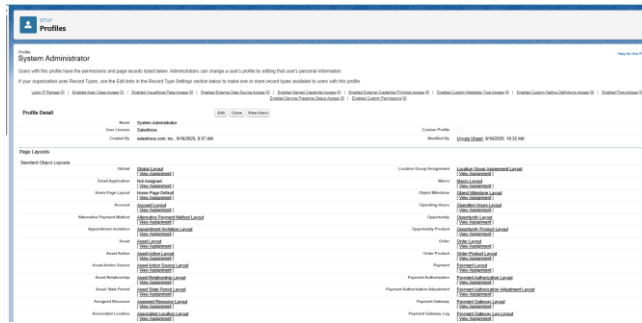
(GMT-07:00) Pacific Daylight Time (America/Los\_Angeles)

Step 3. Business Hours

|           |          |    |          |          |
|-----------|----------|----|----------|----------|
| Sunday    | 12:00 AM | to | 12:00 AM | 24 hours |
| Monday    | 12:00 AM | to | 12:00 AM | 24 hours |
| Tuesday   | 12:00 AM | to | 12:00 AM | 24 hours |
| Wednesday | 12:00 AM | to | 12:00 AM | 24 hours |
| Thursday  | 12:00 AM | to | 12:00 AM | 24 hours |
| Friday    | 12:00 AM | to | 12:00 AM | 24 hours |
| Saturday  | 12:00 AM | to | 12:00 AM | 24 hours |

SaveCancel

- **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).

Project Detail

Standard Buttons

Edit
Delete
Clone
Change Owner
Change Record Type
Printable View
Edit Labels

Custom Buttons

Information (Header visible on edit only)

Project Name

Sample Text

Account Name

Sample Text

Project Description

Sample Text

Opportunity

Sample Text

Details

Project Type

Sample Text

Project Status

Sample Text

Start Date

9/22/2025

End Date

9/22/2025

Hours

Total Hours

542.02

Hours Worked To Date

479.00

Remaining Hours

528.98

Invoicing?

Last Month's Timesheet

Sample Text

System Information (Header visible on edit only)

Created By

Sample Text

Last Modified By

Sample Text

Custom Links (Header visible on edit only)

- **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           | Account_Name__c           | Master-Detail(Account)          |
| Created By             | CreatedById               | Lookup(User)                    |
| End Date               | End_Date__c               | Date                            |
| Hours Worked To Date   | Hours_Worked_To_Date__c   | Roll-Up Summary (SUM Timesheet) |
| Last Modified By       | LastModifiedById          | Lookup(User)                    |
| Last Month's Timesheet | Last_Month_s_Timesheet__c | Formula (Text)                  |
| Opportunity            | Opportunity__c            | Lookup(Opportunity)             |
| Project Description    | Project_Description__c    | Text(255)                       |
| Project Name           | Name                      | Text(80)                        |
| Project Status         | Project_Status__c         | Picklist                        |
| Project Type           | Project_Type__c           | Picklist                        |
| Remaining Hours        | Remaining_Hours__c        | Formula (Number)                |
| Start Date             | Start_Date__c             | Date                            |
| Total Hours            | Total_Hours__c            | Number(16, 2)                   |

## 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.

The screenshot shows the Salesforce Object Manager interface for the 'Project' object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The 'Details' section is selected, showing the following information:

| Details             |                                     |
|---------------------|-------------------------------------|
| Description         | Main Project Management app object  |
| API Name            | Project__c                          |
| Custom              | ✓                                   |
| Singular Label      | Project                             |
| Plural Label        | Projects                            |
| Enable Reports      | ✓                                   |
| Track Activities    | ✓                                   |
| Track Field History |                                     |
| Deployment Status   | Deployed                            |
| Help Settings       | Standard salesforce.com Help Window |

- **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.

The screenshot shows the Salesforce Object Manager interface for the 'Timesheet' object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The 'Details' section is selected, showing the following information:

| Details             |                                     |
|---------------------|-------------------------------------|
| Description         |                                     |
| API Name            | Timesheet__c                        |
| Custom              | ✓                                   |
| Singular Label      | Timesheet                           |
| Plural Label        | Timesheets                          |
| Enable Reports      | ✓                                   |
| Track Activities    | ✓                                   |
| Track Field History |                                     |
| Deployment Status   | Deployed                            |
| Help Settings       | Standard salesforce.com Help Window |

## 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).

| Project                      |                        |                          |                                  |                           |
|------------------------------|------------------------|--------------------------|----------------------------------|---------------------------|
| Details                      | Fields & Relationships |                          |                                  |                           |
| Fields & Relationships       | FIELD LABEL            | FIELD NAME               | DATA TYPE                        | CONTROLLING FIELD INDEXED |
| Page Layouts                 | Account Name           | Account_Name_c           | Master-Detail(Account)           | ✓                         |
| Lightning Record Pages       | Created By             | CreatedById              | Lookup(User)                     |                           |
| Buttons, Links, and Actions  | End Date               | End_Date_c               | Date                             |                           |
| Compact Layouts              | Hours Worked To Date   | Hours_Worked_To_Date_c   | Roll-Up Summary (Date Timesheet) |                           |
| Field Sets                   | Last Modified By       | LastModifiedById         | Lookup(User)                     |                           |
| Object Limits                | Last Month's Timesheet | Last_Month_c_Timesheet_c | Formula (Text)                   |                           |
| Record Types                 | Opportunity            | Opportunity_c            | Lookup(Opportunity)              | ✓                         |
| Related Lookup Filters       | Project Description    | Project_Description_c    | Text(255)                        |                           |
| Search Layouts               | Project Name           | Name                     | Text(255)                        | ✓                         |
| List View Button Layout      | Project Status         | Project_Status_c         | Picklist                         |                           |
| Restriction Rules            | Project Type           | Project_Type_c           | Picklist                         |                           |
| Scoping Rules                | Remaining Hours        | Remaining_Hours_c        | Formula (Number)                 |                           |
| Triggers                     | Start Date             | Start_Date_c             | Date                             |                           |
| Flow Triggers                | Total Hours            | Total_Hours_c            | Number(16, 2)                    |                           |
| Validation Rules             |                        |                          |                                  |                           |
| Conditional Field Formatting |                        |                          |                                  |                           |

### • Timesheet Object Fields:

- Task Date (Date): Indicates when the work was done.
- Hours Worked (Number): Total hours logged (validation ensures  $\leq 8$  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.

| Timesheet                   |                        |                    |                        |                           |
|-----------------------------|------------------------|--------------------|------------------------|---------------------------|
| Details                     | Fields & Relationships |                    |                        |                           |
| Fields & Relationships      | FIELD LABEL            | FIELD NAME         | DATA TYPE              | CONTROLLING FIELD INDEXED |
| Page Layouts                | Consultant             | Consultant_c       | Lookup(User)           | ✓                         |
| Lightning Record Pages      | Created By             | CreatedById        | Lookup(User)           |                           |
| Buttons, Links, and Actions | Invoice?               | invoiced_c         | Checkbox               |                           |
| Compact Layouts             | Last Modified By       | LastModifiedById   | Lookup(User)           |                           |
| Field Sets                  | Project Name           | Project_Name_c     | Master-Detail(Project) | ✓                         |
| Object Limits               | Task Date              | Task_Date_c        | Date                   |                           |
| Record Types                | Task Description       | Task_Description_c | Text(255)              |                           |
| Related Lookup Filters      | Timesheet Number       | Name               | Auto Number            | ✓                         |
| Search Layouts              | Total Hours            | Total_Hours_c      | Number(16, 2)          |                           |
| List View Button Layout     |                        |                    |                        |                           |
| Restriction Rules           |                        |                    |                        |                           |
| Scoping Rules               |                        |                    |                        |                           |

## 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

Account Name

Field Name

Account\_Name

Namespace Prefix

Description

Help Text

Related To

Account

Child Relationship Name

Projects

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

Projects

Reparentable Master Detail

☐ Child records can be reparented to other parent records after they are created

Save

Cancel

- **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:

This text displays on detail and edit pages when users hover over the Info icon next to this field.

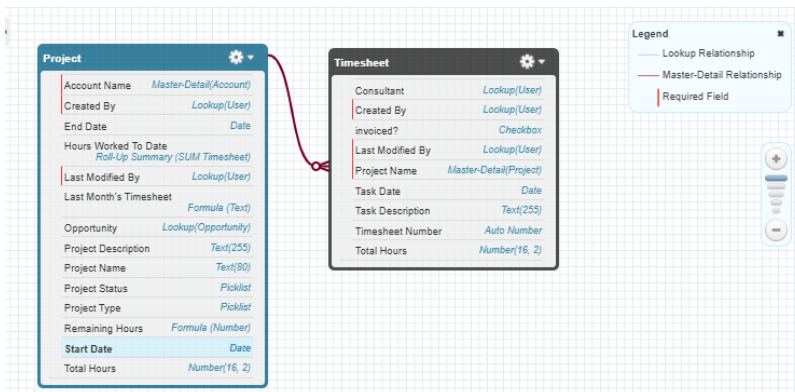
Related To:   
Select the other object to which this object is related.

Child Relationship Name:

Related List Label:   
Specify the title that the related list will have in all of the layouts associated with the parent.

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.

**Edit Hours Worked To Date (Object: Project)**

Field Label:

Field Name:

Namespace Prefix:

Description:

Help Text:

This text displays on detail and edit pages when users hover over the Info icon next to this field.

Calculation Options:   
☐ Force a mass recalculation of this field  
☒ Automatic calculation (Recommended)

Summarized Object:

Summary Type:

Field to Aggregate:

Filter Criteria: Roll-Up Summary field filter criteria can't be edited in Schema Builder. To edit the filter criteria, open the field's edit page.  
☐ Only records meeting certain criteria should be included in the calculation  
☒ All records should be included in the calculation

## 5. Formula Fields

Formula fields provide dynamic calculations:

- **Remaining Hours:**
  - `Total_Hours__c - 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.

The screenshot shows the 'Edit Remaining Hours (Object: Project)' configuration page. It includes fields for 'Field Label' (Remaining Hours), 'Field Name' (Remaining\_Hours), 'Namespace Prefix', 'Description', and 'Help Text'. The 'Return Type' is set to 'Number' and 'Decimal Places' is set to '2'. The 'Formula' section shows the formula `Total_Hours__c - Hours_Worked_To_Date__c`. A 'Functions' sidebar on the right lists various functions like ABS, ACOS, ADDMONTHS, etc. At the bottom, there are options for 'Blank Fields' (Treat blank fields as zeroes or blanks) and 'Save'/'Cancel' buttons.

## 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:

- `Hours_Worked__c > 8`

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

The screenshot shows the 'Timesheet Validation Rule' configuration page. It includes a 'Validation Rule Detail' section with fields for 'Rule Name' (Rstest\_Hours\_per\_Day), 'Error Condition Formula' (`Total_Hours__c > 8`), 'Error Message' (Hours cannot exceed 8 per day), and 'Description' (Hours cannot exceed 8 per day). The 'Active' checkbox is checked. The 'Created By' field shows 'Urvashi Sharma' and the 'Modified By' field shows 'Urvashi Sharma'. There are 'Edit' and 'Close' buttons at the bottom.



New Timesheet

\* Required Information

Information

Project Name  
Third Project X

Consultant  
Uyalla Ushan X

Task Date  
9/27/2025

Task Description

Total Hours  
10.00

Hours cannot be greater than 10.00

Invicid? ☐

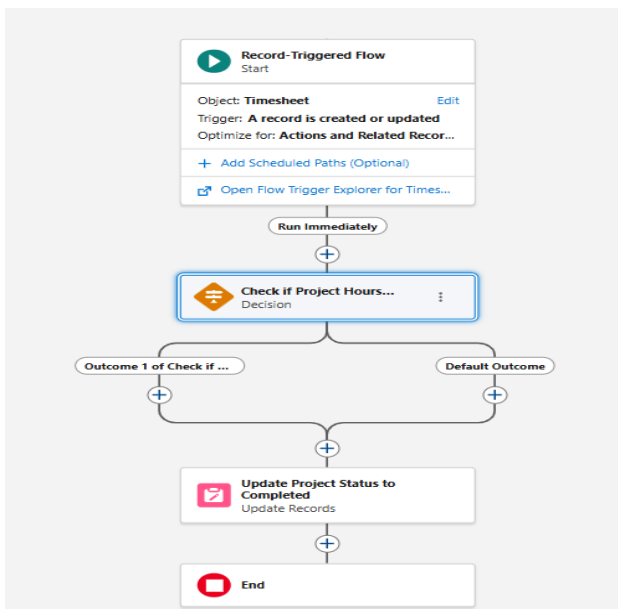
We hit a snag.  
Review the following fields  
Total Hours

Cancel Save & New 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 records 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. Minimizing unnecessary flow to conserve your org's resources.  
If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated** 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 flexible 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 Path ☐

- **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  
Uyyala Uhasri

Task Date  
9/28/2025

Task Description

Total Hours  
8.00

Invoiced?  
☐

Cancel Save & New Save

Project Management Projects Timesheets Reports

Timesheet 2025-10

Timesheet "a026000060M3HAAG" was created.

Related Details

Project Name  
This is my first project

Consultant  
Uyyala Uhasri

Task Date  
9/28/2025

Task Description

Total Hours  
8.00

Invoiced?  
☐

Created By  
Uyyala Uhasri, 9/26/2025, 8:52 AM

Last Modified By  
Uyyala Uhasri, 9/26/2025, 8:52 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.

Project Management

Projects ▼ Timesheets ▼ Reports ▼

Timesheet

2025-1

Related Details

|                  |  |  |
|------------------|--|--|
| Project Name     | <a href="#">This is my first project</a> |  |
| Consultant       |  |  |
| Task Date        | 9/19/2025                                |  |
| Task Description | This is what I did                       |  |
| Total hours      | 6.50                                     |  |
| Involved?        | <input type="checkbox"/>                 |  |

Created By

Uppala Uthappan - 9/19/2025, 11:37 AM

Last Modified By

Uppala Uthappan - 9/19/2025, 11:37 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.

```

1 public class ProjectController {
2
3     @RequestMapping(cacheable=true)
4     public static List<Project_c> fetchProjects() {
5         List<Project_c> projects = [
6             SELECT ID, Name, total_hours_c, hours_worked_to_date_c, status_c
7             FROM Project_c
8             ORDER BY Name
9         ];
10        return projects;
11    }
12
13    @RequestMapping
14    public static void updateProjectStatus(List<Id> projectIds) {
15        List<Project_c> projectToUpdate = [
16            SELECT ID, total_hours_c, hours_worked_to_date_c, status_c
17            FROM Project_c
18            WHERE id IN :projectIds
19        ];
20
21        for(Project_c proj : projectToUpdate){
22            if(proj.hours_worked_to_date_c == proj.total_hours_c){
23                proj.status_c = "Completed";
24            } else if(proj.hours_worked_to_date_c > 0){
25                proj.status_c = "In Progress";
26            } else {
27                proj.status_c = "Not Started";
28            }
29        }
30        update projectToUpdate;
31    }
32
33    @RequestMapping(cacheable=true)
34    public static Map<Id, Decimal> calculateProgress(List<Id> projectIds){
35        Map<Id, Decimal> progressMap = new Map<Id, Decimal>();
36        List<Project_c> projects = [
37            SELECT ID, total_hours_c, hours_worked_to_date_c
38            FROM Project_c
39            WHERE id IN :projectIds
40        ];
41
42        for(Project_c proj : projects){
43            Decimal progress = 0;
44            if(proj.total_hours_c > 0){
45                progress = (proj.hours_worked_to_date_c / proj.total_hours_c) * 100;
46            }
47            progressMap.put(proj.id, progress.setScale(2));
48        }
49        return progressMap;
50    }
51 }

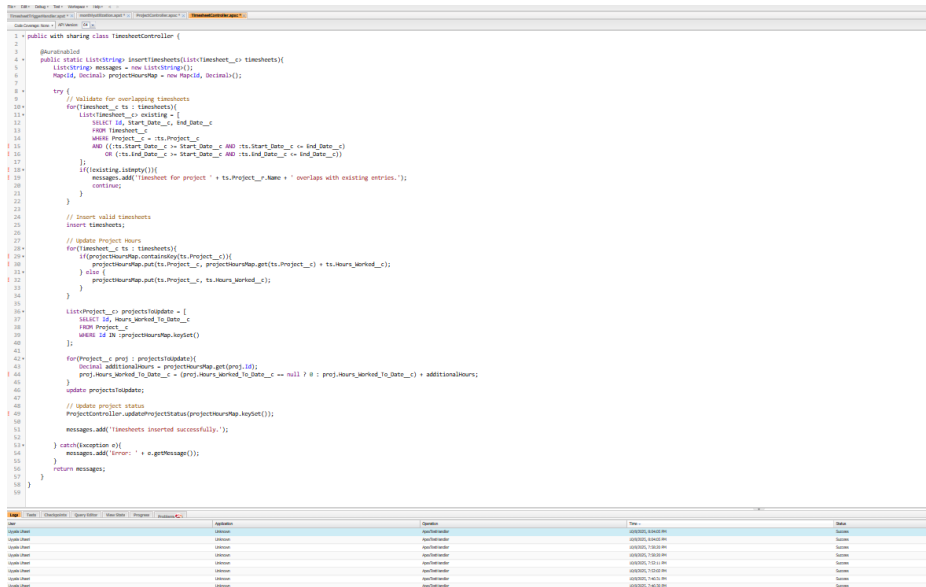
```

The screenshot shows the IntelliJ IDEA IDE interface. The top part displays the code for `ProjectController.java`, which includes methods for fetching projects, updating their status, and calculating their progress. The bottom part shows the 'Run' tab, displaying a table of execution results for five consecutive runs of the application.

| # | Type      | Message  | Status    |
|---|-----------|--|-----------|
| 1 | App Start | Application started. URL: http://localhost:8080/ | Succeeded |
| 2 | App Stop  | Application stopped.                             | Succeeded |
| 3 | App Start | Application started. URL: http://localhost:8080/ | Succeeded |
| 4 | App Stop  | Application stopped.                             | Succeeded |
| 5 | App Start | Application started. URL: http://localhost:8080/ | Succeeded |

- **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.

| NAME       | DESCRIPTION | STATUS    | START DATE          | END DATE            | PROJECT   | PROJECT ID | OPERATOR | TIME     | DATE                |
|------------|-------------|-----------|---------------------|---------------------|-----------|------------|----------|----------|---------------------|
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |

### 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.

| NAME       | DESCRIPTION | STATUS    | START DATE          | END DATE            | PROJECT   | PROJECT ID | OPERATOR | TIME     | DATE                |
|------------|-------------|-----------|---------------------|---------------------|-----------|------------|----------|----------|---------------------|
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |
| Batch Apex | Batch Apex  | Completed | 2023-03-10 10:00:00 | 2023-03-10 10:00:00 | Project A | 1          | System   | 10:00:00 | 2023-03-10 10:00:00 |

- 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.

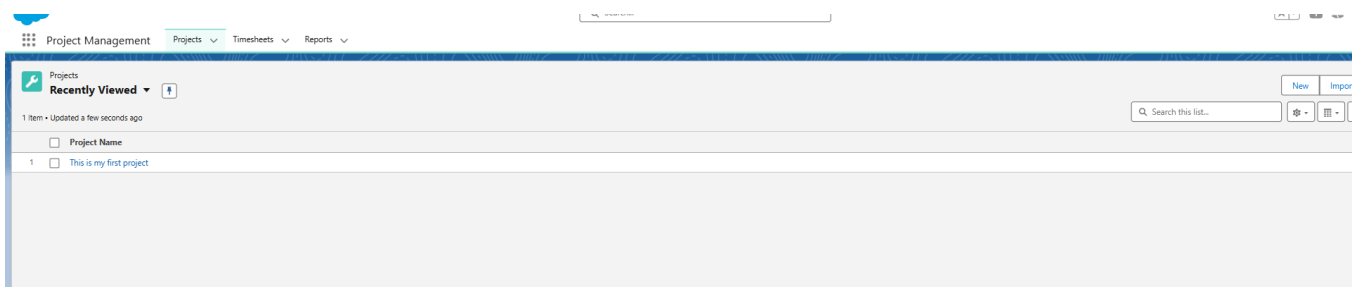
## Phase 6: User Interface Development

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.

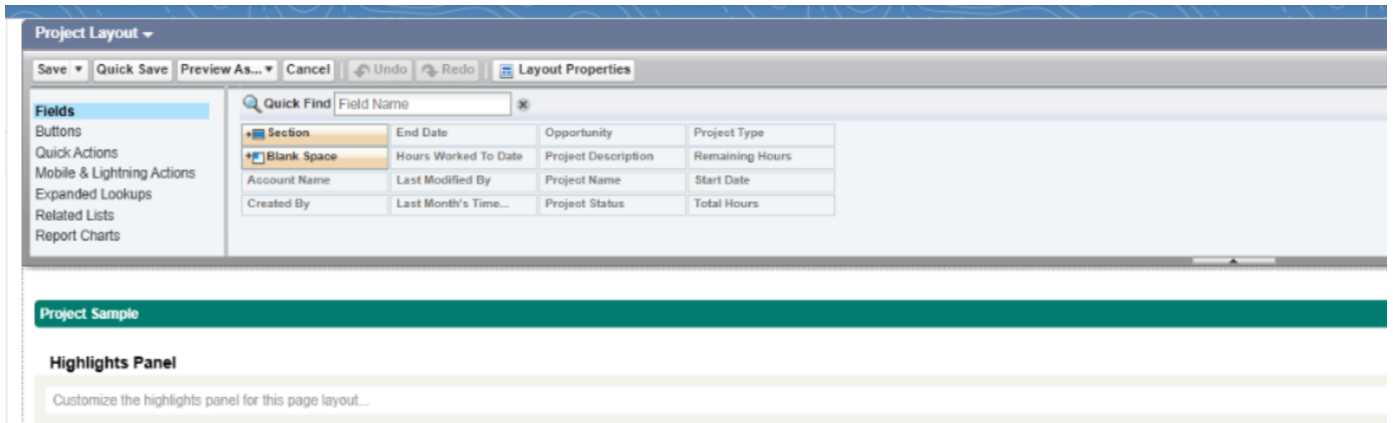
- **Navigation Tabs:**
  - Projects: Displays all active projects and allows easy navigation to individual project records.
  - Timesheets: Shows consultant timesheets with filters for date range and status.
  - Reports & Dashboards: Gives managers quick access to project utilization reports.



- **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.

```
projectList.js
javascript
Copy code

import { LightningElement, track, wire } from 'lwc';
import fetchProjects from '@salesforce/apex/ProjectController.fetchProjects';

const actions = [
  { label: 'View', name: 'view' }
];

const columns = [
  { label: 'Project Name', fieldName: 'Name', type: 'text' },
  { label: 'Status', fieldName: 'Status__c', type: 'text' },
  { label: 'Progress %', fieldName: 'Progress__c', type: 'percent' },
  { type: 'action', typeAttributes: { rowActions: actions } }
];

export default class ProjectList extends LightningElement {
  @track projects = [];
  @track statusFilter = '';
  @track columns = columns;
  statusOptions = [
    { label: 'All', value: '' },
    { label: 'Planned', value: 'Planned' },
    { label: 'In Progress', value: 'In Progress' },
    { label: 'Completed', value: 'Completed' }
  ];

  @wire(fetchProjects)
  wiredProjects({ error, data }) {
    if (data) {
      this.projects = data.map(proj => ({...proj, Progress__c: (proj.Hours_Worked_To_Date__c / proj.Total_Hours_Worked__c) * 100}));
    } else if (error) {
      console.error(error);
    }
  }

  handleStatusChange(event) {
    this.statusFilter = event.detail.value;
    if (this.statusFilter) {
      this.projects = this.projects.filter(p => p.Status__c === this.statusFilter);
    }
  }

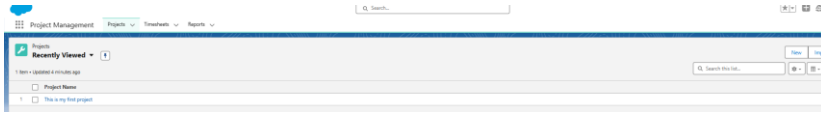
  handleRowAction(event) {
    const actionName = event.detail.action.name;
    const row = event.detail.row;
    if (actionName === 'view') {
      // Navigate to project record
      this[NavigationMixin.Navigate]({
        type: 'standard__recordPage',
        attributes: {
          recordId: row.Id,
          objectApiName: 'Project__c',
          actionName: 'view'
        }
      });
    }
  }
}
```

projectList.html

```
html

<template>
  <lightning-card title="Projects">
    <lightning-combobox
      label="Filter by Status"
      value={statusFilter}
      options={statusOptions}
      onchange={handleStatusChange}>
    </lightning-combobox>

    <lightning-datatable
      key-field="Id"
      data={projects}
      columns={columns}
      onrowaction={handleRowAction}>
    </lightning-datatable>
  </lightning-card>
</template>
```



- **projectDetail:**

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

projectDetail.html

```
html
<template>
  <lightning-card title="Project Detail">
    <p><b>Allocated Hours:</b> {project.Total_Hours__c}</p>
    <p><b>Hours Worked:</b> {project.Hours_Worked_To_Date__c}</p>
    <p><b>Remaining Hours:</b> {remainingHours}</p>

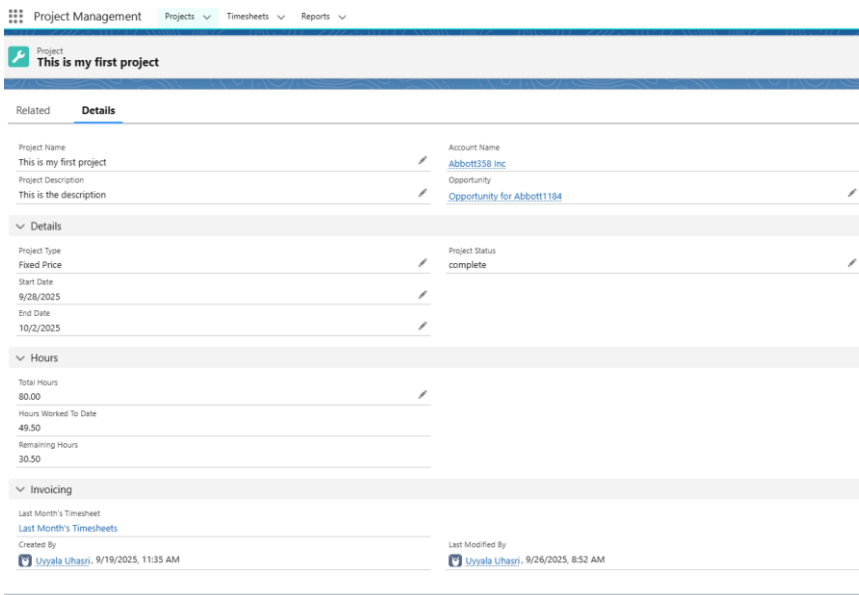
    <lightning-progress-bar value={progress} variant="success"></lightning-progress-bar>
    <p>{progress}% Complete</p>
  </lightning-card>
</template>
```

```
javascript
import { LightningElement, api, wire } from 'lwc';
import fetchProjects from '@salesforce/apex/ProjectController.fetchProjects';

export default class ProjectDetail extends LightningElement {
  @api recordId;
  project;
  progress = 0;
  get remainingHours() {
    return this.project ? this.project.Total_Hours__c - this.project.Hours_Worked_To_Date__c : 0;
  }

  @wire(fetchProjects)
  wiredProject({ error, data }) {
    if(data){
      this.project = data.find(p => p.Id === this.recordId);
      this.progress = this.project ? ((this.project.Hours_Worked_To_Date__c / this.project.Total_Hours__c) * 100) : 0;
    }
  }
}
```

- 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.



timesheetEntry.html

```
html
<template>
  <lightning-card title="Enter Timesheet">
    <lightning-record-edit-form object-api-name="Timesheet__c" onSuccess={handleSuccess}>
      <lightning-messages></lightning-messages>
      <lightning-input-field field-name="Project__c"></lightning-input-field>
      <lightning-input-field field-name="Task_Date__c"></lightning-input-field>
      <lightning-input-field field-name="Hours_Worked__c"></lightning-input-field>
      <lightning-input-field field-name="Description__c"></lightning-input-field>
      <lightning-button type="submit" label="Save" class="slds-m-top_small"></lightning-button>
    </lightning-record-edit-form>
  </lightning-card>
</template>
```

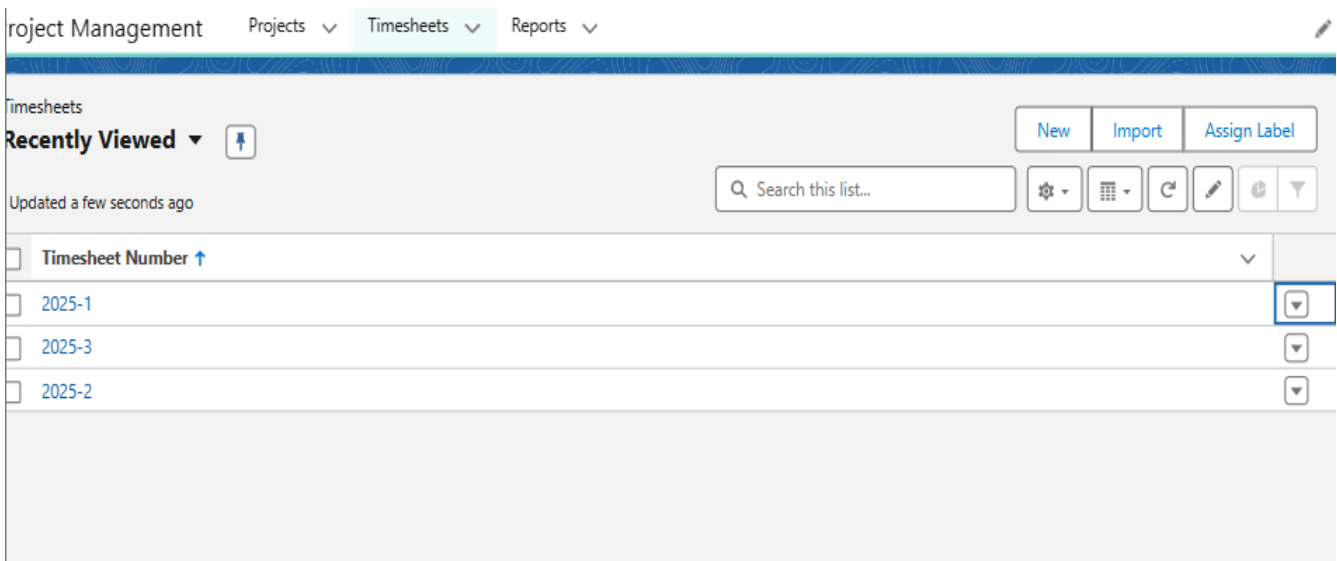
timesheetEntry.js

```
javascript
import { LightningElement } from 'lwc';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';

export default class TimesheetEntry extends LightningElement {
  handleSuccess(event){
    this.dispatchEvent(
      new ShowToastEvent({
        title: 'Success',
        message: 'Timesheet saved successfully',
        variant: 'success'
      })
    );
    // Optional: Refresh project totals using pubsub or refreshApex
  }
}
```

- **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.

```

javascript
Copy code
import { NavigationMixin } from 'lightning/navigation';
export default class SomeComponent extends NavigationMixin(LightningElement) {
  navigateToProject(recordId){
    this[NavigationMixin.Navigate]({
      type: 'standard__recordPage',
      attributes: {
        recordId: recordId,
        objectApiName: 'Project__c',
        actionName: 'view'
      }
    });
  }
}

```

- 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.

| 13 Items • Updated a few seconds ago |                                  | SEARCH THIS LIST... | ⌵ | ⌵ | ⌵ | ⌵ | ⌵ | ⌵ | ⌵ |
|--------------------------------------|----------------------------------|---------------------|---|---|---|---|---|---|---|
| <input type="checkbox"/>             | Timesheet Number                 |                     |   |   |   |   |   |   |   |
| 1                                    | <input type="checkbox"/> 2025-13 |                     |   |   |   |   |   |   | ⌵ |
| 2                                    | <input type="checkbox"/> 2025-12 |                     |   |   |   |   |   |   | ⌵ |
| 3                                    | <input type="checkbox"/> 2025-11 |                     |   |   |   |   |   |   | ⌵ |
| 4                                    | <input type="checkbox"/> 2025-10 |                     |   |   |   |   |   |   | ⌵ |
| 5                                    | <input type="checkbox"/> 2025-9  |                     |   |   |   |   |   |   | ⌵ |
| 6                                    | <input type="checkbox"/> 2025-8  |                     |   |   |   |   |   |   | ⌵ |
| 7                                    | <input type="checkbox"/> 2025-7  |                     |   |   |   |   |   |   | ⌵ |
| 8                                    | <input type="checkbox"/> 2025-6  |                     |   |   |   |   |   |   | ⌵ |

- **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  | Editor         | Created By        | Created On          | Submitted |
|---------------------------------------|--|----------------|-------------------|---------------------|-----------|
| Last Month's Timesheet                |  | Public Reports | Ugarte/Chan       | 9/18/2021, 11:00 AM | Submitted |
| Sample Flow Report Screen Flow        | Which flows run, what's the status of each interface, and how long do users take to complete the screen? | Public Reports | Automated Process | 9/16/2021, 8:57 AM  |           |
| Sample Report Orchestration Run Logs  | What orchestration run logs were created and what happened in their associated orchestration runs?       | Public Reports | Automated Process | 9/16/2021, 8:57 AM  |           |
| Sample Report Orchestration Run       | What orchestration runs have been created and what's the current status of each run?                     | Public Reports | Automated Process | 9/16/2021, 8:57 AM  |           |
| Sample Report Orchestration Stage Run | What orchestration stage runs have been created and what's the current status of each run?               | Public Reports | Automated Process | 9/16/2021, 8:57 AM  |           |
| Sample Report Orchestration Step Run  | What orchestration step runs have been created and what's the current status of each run?                | Public Reports | Automated Process | 9/16/2021, 8:57 AM  |           |
| Sample Report Orchestration Work Item | What orchestration work items were created and what's the current status of each work item?              | Public Reports | Automated Process | 9/16/2021, 8:57 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                                   | Created By        | Created On         | Last Modified By  | Last Modified Date |
|--|-------------------|--------------------|-------------------|--------------------|
| Enstein Bot Reports                    | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |
| Enstein Bot Reports Spring '23         | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |
| Enstein Bot Reports Summer '23         | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |
| Enstein Bot Reports Summer '22         | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |
| Enstein Bot Reports Winter '23         | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |
| Enblement Dashboard Reports Spring '24 | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |
| Enblement Dashboard Reports Summer '24 | Automated Process | 9/16/2025, 6:30 PM | Automated Process | 9/16/2025, 6:30 PM |

## **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).

## 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.

## Sharing Settings

[Help for this Page](#)

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data. Go to [Background Jobs](#) to monitor the progress of a change to an organization-wide default or a parallel sharing recalculation.

Manage sharing settings for: **Project**

[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 |
|---------|-------------------------|-------------------------|--------------------------------|
| Project | Controlled by Parent    | Controlled by Parent    |                                |

#### Other Settings

[Other Settings Help](#)

- Manager Groups ☐ [i](#)
- Secure guest user record access ☒ [i](#)
- Require permission to view record names in lookup fields ☐ [i](#)

### Sharing Rules

You cannot create sharing rules for this item.

### Sharing Overrides

#### Profiles That Override Parent's Sharing

[Sharing Overrides Help](#)

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

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 ☐ [i](#)
- Secure guest user record access ☒ [i](#)
- Require permission to view record names in lookup fields ☐ [i](#)

### Sharing Rules

You cannot create sharing rules for this item.

### Sharing Overrides

#### Profiles That Override Parent's Sharing

[Sharing Overrides Help](#)

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                  |
| <a href="#">Analytics Cloud Integration User</a> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| <a href="#">System Administrator</a>             | <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

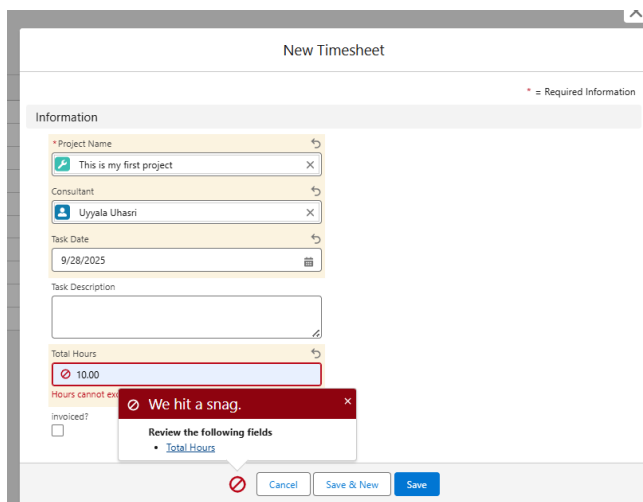
- Scope of Testing:
  - Custom Objects (Projects, Timesheets)
  - Validation Rules
  - Record-Triggered Flows
  - Approval Processes
  - Apex Triggers & Batch Jobs
  - Reports and Dashboards
- Testing Methodology:
  - Manual testing for record creation, validation rules, flows, and approvals.
  - Automated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



The screenshot shows a 'New Timesheet' form in Salesforce. The form includes the following fields and values:

- Project Name:** This is my first project
- Consultant:** Uyyala Uhasri
- Task Date:** 9/28/2025
- Task Description:** (Empty)
- Total Hours:** 10.00 (This field is highlighted in red, indicating an error)
- Invoiced?:** ☐

A red error message box is displayed over the Total Hours field, stating: "We hit a snag. Review the following fields: Total Hours". At the bottom of the form, there are three buttons: "Cancel", "Save & New", and "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.

**Project Third Project**

Related **Details**

Project Name: Third Project | Account Name: Abbott358 Inc

Project Description: This is my third project | Opportunity: Opportunity for Abbott1184

**Details**

Project Type: Fixed Price | Project Status: ongoing

Start Date: 9/26/2025

End Date: 9/30/2025

**Hours**

Total Hours: 55.00

Hours Worked To Date: 33.00

Remaining Hours: 22.00

**Invoicing**

Last Month's Timesheet: Last Month's Timesheets

Created By: Uyyala Uhasri, 9/25/2025, 11:18 AM | Last Modified By: Uyyala Uhasri, 9/25/2025, 11:18 AM

**Project Third Project**

Related **Details**

Project Name: Third Project | Account Name: Abbott358 Inc

Project Description: This is my third project | Opportunity: Opportunity for Abbott1184

**Details**

Project Type: Fixed Price | Project Status: ongoing

Start Date: 9/26/2025

End Date: 9/30/2025

**Hours**

Total Hours: 55.00

Hours Worked To Date: 33.00

Remaining Hours: 22.00

**Invoicing**

Last Month's Timesheet: Last Month's Timesheets

Created By: Uyyala Uhasri, 9/25/2025, 11:18 AM | Last Modified By: Uyyala Uhasri, 9/25/2025, 11:18 AM

### 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.