

Workflow Assignment

Basic Assignments

1. **Objective:** Create "Department" and "Employee" windows to manage department and employee information.
 - a. **Task Details:**
 - i. **Department Window Design:** Fields required:
 1. Department ID (Text - required)
 2. Department Name (Text - required)
 3. Location (Text - optional)
 - ii. **Employee Window Design:** Fields required:
 1. Employee ID (Text - required)
 2. Name (Text - required)
 3. Department (Foreign Key referencing Department - required)
 4. Position (Text - optional)
 - iii. Provide clear labels and placeholder text.
 - b. **Deliverables:** "Department" and "Employee" windows with structured designs and appropriate table views.
2. **Objective:** Design "Project" and "Team Member" windows.
 - a. **Task Details:**
 - i. **Project Window Design:** Fields required:
 1. Project Name (Text - required)
 2. Start Date (Date - required)
 3. End Date (Date - optional)
 4. Project Manager (Text - optional)
 - ii. **Team Member Window Design:** Fields required:
 1. Team Member ID (Text - required)
 2. Name (Text - required)
 3. Assigned Project (Foreign Key referencing Project - required)
 4. Role (Enum - options like Developer, Designer, Tester)
 - b. **Deliverables:** Completed "Project" and "Team Member" windows with user-friendly input fields and appropriate table views.
3. **Objective:** Create "Student" and "Course Enrollment" windows for tracking student registrations.
 - a. **Task Details:**
 - i. **Student Window Design:** Fields required:
 1. Student ID (Text - required)

2. Name (Text - required)
3. Program (Enum - options like Science, Arts, Commerce)
- ii. **Course Enrollment Window Design:** Fields required:
 1. Enrollment ID (Auto-generated/Sequence - read-only)
 2. Student ID (Foreign Key referencing Student - required)
 3. Course (Text - required)
 4. Enrollment Date (Date - required)
- b. **Deliverables:** User-friendly "Student" and "Course Enrollment" windows with appropriate table views.

Intermediate Assignments

1. **Objective:** Design "Expense" and "Expense Report" windows with email notifications.
 - a. **Task Details:**
 - i. **Expense Window Design:** Fields required:
 1. Expense ID (Text - required)
 2. Category (Enum - required, options like Travel, Meals, Supplies)
 3. Amount (Float - required)
 4. Date (Date - required)
 - ii. **Expense Report Window Design:** Fields required:
 1. Report ID (Auto-generated/Sequence - read-only)
 2. Employee ID (Lookup - required)
 3. Total Expenses (Custom Formula- disabled)
 4. Report Date (Date - required)
 - iii. **Workflow Requirement:** Send a summary email of total expenses upon report creation.
 - b. **Deliverables:** Functional "Expense" and "Expense Report" windows with email notifications.
2. **Objective:** Build "Inventory" and "Restock Request" windows with a low-stock alert workflow.
 - a. **Task Details:**
 - i. **Inventory Window Design:** Fields required:
 1. Product ID (Text - required)
 2. Product Name (Text - required)
 3. Stock Quantity (Integer - required)
 4. Reorder Level (Integer - required)
 - ii. **Restock Request Window Design:** Fields required:
 1. Request ID (Auto-generated/Sequence - read-only)

2. Product ID (Foreign Key referencing Inventory - required)
 3. Quantity Requested (Integer - required)
 4. Request Date (Date - required)
 - iii. **Workflow Requirement:** Trigger a low-stock alert when stock falls below the reorder level.
 - b. **Deliverables:** "Inventory" and "Restock Request" windows with automated low-stock alerts.
3. **Objective:** Develop "Task" and "Task Assignment" windows with a workflow to notify assigned members.
- a. **Task Details:**
 - i. **Task Window Design:** Fields required:
 1. Task ID (Auto-generated/Sequence - read-only)
 2. Task Name (Text - required)
 3. Due Date (Date - required)
 4. Status (Enum - options like Not Started, In Progress, Completed)
 - ii. **Task Assignment Window Design:** Fields required:
 1. Assignment ID (Auto-generated/Sequence - read-only)
 2. Task ID (Foreign Key referencing Task - required)
 3. Assigned To (User Lookup - required)
 4. Assignment Date (Date - required)
 - iii. **Workflow Requirement:** Send an email notification to assigned members upon task assignment.
 - b. **Deliverables:** "Task" and "Task Assignment" windows with email notification workflow.

Advanced Assignments

1. **Objective:** Create "Invoice" and "Payment" windows with an automatic balance update.
 - a. **Task Details:**
 - i. **Invoice Window Design:** Fields required:
 1. Invoice ID (Auto-generated/Sequence - read-only)
 2. Customer Name (Text - required)
 3. Invoice Amount (Float - required)
 4. Status (Enum - required, options like Unpaid, Paid, Overdue)
 - ii. **Payment Window Design:** Fields required:
 1. Payment ID (Auto-generated/Sequence - read-only)
 2. Invoice ID (Foreign Key referencing Invoice - required)

3. Payment Amount (Float - required)
 4. Payment Date (Date - required)
 - iii. **Workflow Requirement:** Deduct the payment amount from the invoice balance upon each payment.
 - b. **Deliverables:** "Invoice" and "Payment" windows with balance update workflow.
2. **Objective:** Design "Client" and "Project" windows with a workflow to calculate project budgets.
- a. **Task Details:**
 - i. **Client Window Design:** Fields required:
 1. Client ID (Auto-generated/Sequence - read-only)
 2. Client Name (Text - required)
 3. Contact Information (Text - required)
 - ii. **Project Window Design:** Fields required:
 1. Project ID (Auto-generated/Sequence - read-only)
 2. Client (Foreign Key referencing Client - required)
 3. Project Budget (Float - required)
 4. Estimated Hours (Integer - required)
 5. Hourly Rate (Float - required)
 6. Total Project Cost (disabled)
 - iii. **Workflow Requirement:** Calculate and update "Total Project Cost" based on "Hourly Rate" and "Estimated Hours."
 - b. **Deliverables:** "Client" and "Project" windows with dynamic project cost calculation workflow.
3. **Objective:** Create "Student" and "Exam" windows with automated score aggregation.
- a. **Task Details:**
 - i. **Student Window Design:** Fields required:
 1. Student ID (Text - required)
 2. Name (Text - required)
 3. Program (Text - required)
 4. Total Score (Float – disabled)
 - ii. **Exam Window Design:** Fields required:
 1. Exam ID (Auto-generated /Sequence - read-only)
 2. Student ID (Foreign Key referencing Student - required)
 3. Subject (Text - required)
 4. Score (Float - required)
 - iii. **Workflow Requirement:** Automatically calculate and display total and average scores for each student across exams.
 - b. **Deliverables:** "Student" and "Exam" windows with automated score aggregation.

4. **Objective:** Design "Employee" and "Performance Review" windows with a workflow for average rating calculation.
- a. **Task Details:**
 - i. **Employee Window Design:** Fields required:
 1. Employee ID (Text - required)
 2. Name (Text - required)
 3. Department (Enum - options like HR, Engineering, Sales)
 - ii. **Performance Review Window Design:** Fields required:
 1. Review ID (Auto-generated/Sequence - read-only)
 2. Employee ID (Foreign Key referencing Employee - required)
 3. Review Period (Text - required)
 4. Rating (Integer - required, range 1-5)
 - iii. **Workflow Requirement:** Calculate the average performance rating per employee across all reviews.
 - b. **Deliverables:** "Employee" and "Performance Review" windows with average rating workflow.

Expert Assignments

5. **Objective:** Create an intuitive and visually appealing interface under the menu "Assignment One" that includes windows for "Course" and "Module" with user-friendly designs, appropriate icons, and clear table views. Additionally, implement a workflow that updates course credits dynamically based on module additions.

Task Details:

- **Course Window Design:**
 - Create a form-based interface for the "Course" with the fields specified:
 - ♣ **Course Name** (Text - required)
 - ♣ **Course Banner** (Image - optional)
 - ♣ **Description** (Text Area - optional)
 - ♣ **Category** (Enum - required, options like Science, Arts, Technology)
 - ♣ **Level** (Enum - required, options like Beginner, Intermediate, Advanced)
 - ♣ **Credits** (Float - disabled)
 - ♣ **Language** (Enum - default to English)
 - ♣ **Prerequisites** (Text Editor - optional)

- o Add relevant information next to input field wherever required to make the design engaging and user-friendly.
- **Module Window Design:**
 - o Design a similar form-based interface for "Module" with these fields:
 - ♣ **Module Name** (Text - required)
 - ♣ **Course** (Foreign Key referencing the Course - required)
 - ♣ **Module Banner** (Image - optional)
 - ♣ **Description** (Text Area - optional)
 - ♣ **Credit** (Float - required; should have validation between 60 and 100)
 - ♣ **Module Type** (Enum - required, options like Lecture, Workshop, Quiz)
 - ♣ **Resource** (File Upload - optional)
 - o Ensure a clean, organized layout.
- **Workflow Requirement:**
 - o Set up a workflow where, upon adding a new module, the selected course's credits are automatically incremented by the credit value assigned to the module.
 - o Ensure error handling if the module's credit value exceeds validation limits or if the course has insufficient remaining credit capacity.

Deliverables:

- Completed "Course" and "Module" windows under the "Assignment One" menu, with a consistent and attractive design.
 - Workflow to dynamically update course credits with new module additions, along with documentation detailing the logic behind this update process.
6. **Objective:** Develop a responsive and user-friendly interface under the "Banking Assignments" menu. This interface should include two windows: "Bank Account" and "Transfer Balance." Implement a workflow to handle balance transfers, automatically updating balances for both the sender and recipient accounts. Additionally, include logic to apply a 1% transaction fee to each transfer, deducting this fee from the sender's account and adding it to a designated bank account record for internal tracking.

Task Details:

- **Bank Account Window Design:**

- Create a structured form-based interface for "Bank Account" with the following fields:
 - ♣ **Account Name** (Text - required)
 - ♣ **Account Number** (Text - required, with validation to ensure uniqueness)
 - ♣ **Account Type** (Enum - required, options like Savings, Checking, Business)
 - ♣ **Balance** (Float - required, with default value set to zero)
 - ♣ **Status** (Enum - required, options like Active, Inactive)
- Add relevant information next to input field wherever required to make the design engaging and user-friendly.
-
- **Transfer Balance Window Design:**
 - Design a similar form-based interface for "Module" with these fields:
 - ♣ **Module Name** (Text - required)
 - ♣ **Course** (Foreign Key referencing the Course - required)
 - ♣ **Module Banner** (Image - optional)
 - ♣ **Description** (Text Area - optional)
 - ♣ **Credit** (Float - required; should have validation between 60 and 100)
 - ♣ **Module Type** (Enum - required, options like Lecture, Workshop, Quiz)
 - ♣ **Resource** (File Upload - optional)
 - Ensure that this form is cleanly laid out, with validations in place to prevent self-transfers or transfers between inactive accounts.
- **Workflow Requirement:** Establish a workflow to handle balance transfers with the following logic:
 - **Check Balance:** Validate if the "From Account" has enough funds to cover the **Transfer Amount + 1% Transaction Fee**.
 - **Transfer Process:**
 - ♣ **Deduct** the transfer amount + 1% transaction fee from the "From Account" balance.
 - ♣ **Credit** the "To Account" balance with the transfer amount (no fee deduction here).
 - **Transaction Fee Handling:**
 - ♣ The deducted 1% transaction fee should automatically be added to a designated bank account created by the bank for internal purposes.
 - **Error Handling:**

- ♣ Ensure clear error messages if:
- ♣ The "From Account" has insufficient funds.
- ♣ Invalid accounts are selected (e.g., inactive accounts or the same account for both sender and receiver).

Deliverables:

- Completed "**Bank Account**" and "**Transfer Balance**" windows under the "Banking Assignments" menu with consistent, visually appealing design.
- **Workflow Implementation** to update account balances, apply transaction fees, and record internal bank credits.
- **Documentation** of the workflow logic, including examples of potential scenarios, error handling procedures, and any assumptions made.

View Manager

Objective

The objective of this assignment is to allow learners to independently design, configure, and enhance any window of their choice using the Dalfin View Manager and Action Builder.

Learners must demonstrate their understanding of:

View configuration

- Action setup (table-level, row-level, and UI actions)
- Views setup
- Validation and user experience best practices

Assignment Overview

Each learner must select one business window of their choice (examples below) and fully configure it with multiple views, actions, and at least one workflow.

Example Window Choices (Learner may choose ONE):

- Employee
- Project
- Task
- Student
- Course
- Inventory
- Invoice
- Bank Account
- Expense
- Any other relevant business entity

Requirements

1. Window & Field Design

Mandatory Field Requirements

Each selected window must contain:

Minimum 5 fields

At least:

- 2 required fields
- 1 optional field
- 1 enum or lookup field
- 1 system or calculated field (disabled / read-only)

2. View Configuration (Minimum 5 Views Required)

The learner must configure at least five (5) different views from the list below:

Available View Types

- Grid
- Table
- List
- Card
- Kanban
- Calendar
- Timeline
- Gantt
- Map
- Accordion
- Tabbed
- Column
- Masonry
- Activity
- Team
- Gridlines

View Expectations

For each view:

- Purpose must be clearly defined
- Fields displayed should be customized
- Sorting, grouping, or filtering must be applied where relevant

3. Action Configuration

A. Table-Level Actions (Minimum 3)

Examples:

- Add Record
- Import Data

- Export Data
- Generate Invoice
- Show Summary Card
- Print
- Bulk Update

B. Row-Level Actions (Minimum 5)

Examples:

- View
- Edit
- Duplicate
- Add Child Record
- Generate QR Code
- Verify
- Share
- Comment
- Highlight

C. Other UI Actions (Minimum 3)

Examples:

- Draft Mode
- Show / Hide Fields
- Show Enumerations as Chips
- Conditional Highlighting
- Status Toggle

Business Intelligence Module

BASIC BI ASSIGNMENTS

1. Department & Employee BI Analysis

Objective: Explore relationships and basic reporting.

Tasks:

1. Create a **Data Package** using:
 - a. Department
 - b. Employee
2. Include fields:
 - a. Department Name
 - b. Location
 - c. Employee ID
 - d. Employee Name
 - e. Position
3. Create the following charts:
 - a. Number of employees per department
 - b. Employees grouped by location
4. Create a **Dashboard**:
 - a. Add both charts
 - b. Apply a department-level filter

Outcome: User understands FK relationships and basic aggregation.

2. Project & Team Member BI Overview

Objective: Learn chart selection and grouping.

Tasks:

1. Create a data package using:
 - a. Project
 - b. Team Member
2. Build charts:
 - a. Team members per project
 - b. Team members by role (Developer, Designer, Tester)
3. Try changing chart types (Bar → Pie → Table)
4. Add a dashboard filter for **Project Name**

Outcome: User experiments with visualization types.

3. Student & Course Enrollment BI

Objective: Practice date filters and record tracking.

Tasks:

1. Create a data package using:
 - a. Student
 - b. Course Enrollment
2. Create charts:
 - a. Total enrollments per course
 - b. Enrollments by program (Science, Arts, Commerce)
3. Apply a **date filter** on Enrollment Date
4. Save a dashboard named:

Student Enrollment Overview

4. INTERMEDIATE BI ASSIGNMENTS

Expense & Expense Report BI

Objective: Learn calculated insights and summaries.

Tasks:

1. Create a data package using:
 - a. Expense
 - b. Expense Report
2. Create charts:
 - a. Total expenses by category
 - b. Expenses by employee
3. Add a **summary KPI chart**:
 - a. Total expenses across all reports
4. Create a dashboard with:
 - a. Category-wise expense
 - b. Employee-wise expense
 - c. Date filter

5. Inventory & Restock BI Monitoring

Objective: Identify alerts and thresholds visually.

Tasks:

1. Create a data package using:
 - a. Inventory
 - b. Restock Request
2. Create charts:
 - a. Products below reorder level
 - b. Stock quantity vs reorder level
3. Highlight low-stock items using filters
4. Create a dashboard named: Inventory Health Dashboard

6. Task & Task Assignment BI

Objective: Track workflow status visually.

Tasks:

1. Create a data package using:
 - a. Task
 - b. Task Assignment
2. Create charts:
 - a. Tasks by status
 - b. Tasks assigned per user
3. Add filters:
 - a. Status
 - b. Due Date
4. Create a dashboard for **Task Monitoring**

ADVANCED BI ASSIGNMENTS

7. Invoice & Payment BI Tracking

Objective: Financial tracking and trend analysis.

Tasks:

1. Create a data package using:
 - a. Invoice
 - b. Payment
2. Create charts:
 - a. Paid vs unpaid invoices
 - b. Total payments received per customer
3. Add a date-based payment trend chart
4. Create a dashboard with Invoice & Payment Summary

8. Client & Project Cost BI

Objective: Cost analysis and performance insights.

Tasks:

1. Create a data package using:
 - a. Client
 - b. Project
2. Create charts:
 - a. Total project cost per client
 - b. Estimated hours vs total project cost
3. Add filters:
 - a. Client
 - b. Project
4. Create a dashboard focused on **Cost Monitoring**

9. Student & Exam Performance BI

Objective: Aggregation and performance comparison.

Tasks:

1. Create a data package using:
 - a. Student
 - b. Exam
2. Create charts:
 - a. Average score per student
 - b. Subject-wise performance
3. Identify top 5 and bottom 5 students using sorting
4. Create a **Performance Dashboard**

10. Employee Performance BI

Objective: Multi-record aggregation.

Tasks:

1. Create a data package using:
 - a. Employee
 - b. Performance Review
2. Create charts:
 - a. Average rating per employee

- b. Department-wise performance
- 3. Apply filters:
 - a. Review Period
 - b. Department
- 4. Save a dashboard with all charts.

EXPERT BI ASSIGNMENTS

11. Course & Module BI (Assignment One Menu)

Objective: Track academic structure & credit distribution.

Tasks:

- 1. Create a data package using:
 - a. Course
 - b. Module
- 2. Create charts:
 - a. Total credits per course
 - b. Modules by module type
- 3. Create dashboard:
 - a. Course overview
 - b. Credit distribution
- 4. Apply course-level filters

12. Banking BI – Account & Transfer Analysis

Objective: Financial flow and audit tracking.

Tasks:

- 1. Create a data package using:
 - a. Bank Account
 - b. Transfer Balance
- 2. Create charts:
 - a. Account balance distribution
 - b. Transfers per account
 - c. Total transaction fees collected
- 3. Create dashboards:
 - a. Banking Overview
 - b. Transaction Monitoring
- 4. Use filters:

- a. Account Type
- b. Date
- c. Status