**📌 Volunteer Management App - Objects & Fields**

| **Object Name** | **Field Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| **Volunteer** | Volunteer Name | Text (Required) | Full name of the volunteer |
|  | Email | Email | Volunteer’s email address |
|  | Phone | Phone | Contact number |
|  | Availability | Picklist | Options: Weekdays, Weekends, Anytime |
|  | Skills | Multi-Select Picklist | List of volunteer skills |
|  | Status | Picklist | Options: Active, Inactive, Pending |
|  | Notes | Long Text Area | Additional information |

| **Event** | Event Name | Text (Required) | Name of the volunteer event | | | Date | Date | Event date | | | Location | Text | Event location | | | Description | Long Text Area | Event details | | | Volunteers Needed | Number | How many volunteers are required? |

**📌 Volunteer Assignment Object & Fields**

The **Volunteer Assignment** object is a **junction object** that connects **Volunteers** and **Events**. This allows a single volunteer to be assigned to multiple events and a single event to have multiple volunteers.

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **Volunteer** | Lookup (Volunteer) | Links to a specific Volunteer record |
| **Event** | Lookup (Event) | Links to a specific Event record |
| **Role** | Picklist | Defines the role of the volunteer in the event (e.g., Organizer, Assistant, Participant) |
| **Status** | Picklist | Options: Assigned, Confirmed, Cancelled (Tracks volunteer status for the event) |
| **Hours Worked** | Number | Tracks the total hours a volunteer has worked for an event |

**Should You Create Lookup Fields in the Volunteer & Event Objects?**

❌ **No, you don’t need lookup fields in the Volunteer or Event objects themselves.**  
✅ Instead, the **Volunteer Assignment** object will act as a **junction object**, linking **Volunteers** and **Events** using Lookup relationships.

**What You Need to Do Now:**

1️⃣ **Go to the Volunteer Assignment Object.**  
2️⃣ **Create Two Lookup Fields:**

* **Volunteer (Lookup to Volunteer)**
* **Event (Lookup to Event)**

**How It Works:**

* Each **Volunteer Assignment** record will link one **Volunteer** to one **Event**.
* This way, you can have multiple volunteers in an event and also allow a single volunteer to participate in multiple events.

✅ **Final Result:**

* **A Volunteer can be assigned to multiple Events.**
* **An Event can have multiple Volunteers assigned.**
* **Volunteer Assignment connects them both**

**✅ Next Steps: Add UI Enhancements & Automations**

Now that the core objects and relationships are set up, let's make it more user-friendly and automate some processes.

**1️⃣ Add Related Lists in Page Layouts**

Since the relationships are working, ensure they are visible properly:

1. **Go to Setup → Object Manager → Volunteer.**
2. **Click on Page Layouts** → Select the default layout.
3. Scroll to **Related Lists** → Ensure **Volunteer Assignments** is added.
4. Click **Save**.
5. Repeat the same steps for **Event Object** and ensure **Volunteer Assignments** appears in its related lists.

📌 **This ensures that users can quickly see all assignments related to a volunteer or an event.**

**2️⃣ Add a Simple Automation using Flow**

Let’s create a simple **automation** that updates the **Volunteer Status** to **"Active"** when they get assigned to an event.

📌 **Steps to Create an Auto-Update Flow:**

1. **Go to Setup** → Search for **"Flows"**.
2. Click **New Flow** → Select **Record-Triggered Flow**.
3. **Choose the Volunteer Assignment object** (since we want to trigger when a new assignment is created).
4. **Set the Trigger Conditions:**
   * Entry condition: **When a record is created.**
5. **Add an "Update Records" Element:**
   * Find the related **Volunteer** (lookup field).
   * Set **Volunteer Status = "Active"**.
6. **Save the Flow** → Name it **"Auto-Activate Volunteers"**.
7. **Activate the Flow.**

📌 **Test This:**

* Create a new **Volunteer Assignment** for a Volunteer who is currently **"Pending"**.
* Check if their status **automatically updates to "Active".**

Clear explanation of flows:

**📌 Step-by-Step: Configure the Update Records Element**

**1️⃣ Click on the "Update Records" Element**

* You should already have added an **Update Records** element in your Flow.
* Click on it to configure it.

**2️⃣ Name the Update Records Element**

* Give it a name like **"Update Volunteer Status"** (this is just for reference).

**3️⃣ Choose "How to Find Records to Update"**

* Select **"Specify conditions to identify records, and set fields individually"** (4th option).
* This allows us to find the Volunteer based on the Volunteer Assignment record.

**4️⃣ Set Filter Conditions to Find the Related Volunteer**

* Click **"Add Condition"** and set it up like this:

| **Field** | **Operator** | **Value** |
| --- | --- | --- |
| Volunteer Assignment → Volunteer | Equals | {!$Record.VolunteerId} |

✅ This ensures we update the **Volunteer record linked to the Volunteer Assignment**.

**5️⃣ Set the Field to Update**

* Under **Set Field Values for the Volunteer Record**, click **"Add Field"**.
* In **Field**, search for **"Status"**.
* In **Value**, type "Active" (without quotes).

📌 This means when a Volunteer is assigned to an Event, their **Status field** will be updated to **Active**.

**6️⃣ Save & Activate the Flow**

* Click **Done** to save the changes.
* Click **Save** and name your Flow **"Auto-Activate Volunteers"**.
* Click **Activate** to enable the Flow.

**✅ Test the Flow**

1. Go to **Volunteer Assignments** and create a new record.
   * Choose a **Volunteer** and an **Event**.
   * Set **any role and status**.
   * Click **Save**.
2. Now, **go to the Volunteer record you assigned**.
3. Check if their **Status is automatically updated to "Active"**.

Important

**📌 Understanding the Flow: Auto-Update Volunteer Status**

**🚀 Goal of the Flow**

When a **Volunteer is assigned to an Event** (i.e., a **Volunteer Assignment record is created**), the **Volunteer’s status should automatically update to "Active"**.

**🛠 How the Flow Works (Simplified Explanation)**

1️⃣ **Flow gets triggered when a new Volunteer Assignment record is created.**

* This means whenever a Volunteer is assigned to an Event, this Flow runs automatically.

2️⃣ **Find the Volunteer connected to this new Volunteer Assignment record.**

* The **Volunteer Assignment** object has a **lookup field** to the **Volunteer object**.
* The Flow identifies the specific **Volunteer related to this assignment**.

3️⃣ **Update the Volunteer’s Status field to "Active".**

* The Flow **updates the "Status" field** of the Volunteer record.
* This ensures that any assigned Volunteer is automatically marked as **Active**.

**📌 Step-by-Step Breakdown of Each Element in the Flow**

**1️⃣ Start Element: "Record-Triggered Flow"**

* **Object:** Volunteer Assignment
* **Trigger Condition:** **"When a record is created"**
* **Why?** Because we want the Flow to run only when a new **Volunteer Assignment** is added.

**2️⃣ Update Records: "Update Volunteer Status"**

* **What it does:** Finds the **Volunteer related to the Volunteer Assignment** and updates their **Status** to "Active".
* **How?**
  + Finds the **Volunteer record where ID matches the Volunteer field in the Volunteer Assignment**.
  + Updates the **Status field** of that Volunteer record to "Active".

**🚀 Real-World Example (How This Works in Action)**

1️⃣ **Before the Flow:**

* You have a Volunteer named **Pavan**, but their Status is "Pending".
* You also have an Event called **"College Intern Program"**.
* You manually create a **Volunteer Assignment**, linking **Pavan** to the **College Intern Program** event.

2️⃣ **Flow Gets Triggered:**

* As soon as you save the **Volunteer Assignment**, Salesforce **automatically runs the Flow**.

3️⃣ **Flow Finds the Related Volunteer Record:**

* The Flow looks at the **Volunteer Assignment** you just created.
* It finds the **Volunteer linked to this assignment (Pavan)**.

4️⃣ **Flow Updates the Volunteer Status:**

* The Flow updates **Pavan’s** Status to "Active".

5️⃣ **After the Flow Runs:**

* Now, when you check Pavan’s Volunteer record, their **Status is automatically set to "Active"**, without you having to update it manually! 🚀

**💡 Why This Flow is Important?**

* It **automates** a manual task (no need to update Volunteer Status manually).
* It **ensures data consistency** (all assigned Volunteers are correctly marked as "Active").
* It **saves time** when working with many Volunteers and Events.

**🔍 Summary: What You Should Remember**

* **Trigger:** The Flow runs when a **Volunteer Assignment** is created.
* **Lookup:** It finds the **Volunteer linked to that assignment**.
* **Update:** It **changes the Status of the Volunteer** to "Active".

**📌 Summary of the Volunteer Management App Project**

**✅ Step 1: Created the Volunteer Management App**

* Used **App Manager** to create a **custom app** named **Volunteer Management**.
* Added **three custom objects**:
  1. **Volunteer**
  2. **Event**
  3. **Volunteer Assignment**
* Configured **tabs for Volunteers, Events, and Volunteer Assignments** so that only **you (Admin) and Sales Reps** can see them.

**✅ Step 2: Created Custom Objects & Fields**

| **Object Name** | **Field Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| **Volunteer** | Volunteer Name | Text (Required) | Full name of the volunteer |
|  | Email | Email | Volunteer’s email address |
|  | Phone | Phone | Contact number |
|  | Availability | Picklist | Options: Weekdays, Weekends, Anytime |
|  | Skills | Multi-Select Picklist | List of volunteer skills |
|  | Status | Picklist | Options: Active, Inactive, Pending |
|  | Notes | Long Text Area | Additional information |

| **Event** | Event Name | Text (Required) | Name of the volunteer event | | | Date | Date | Event date | | | Location | Text | Event location | | | Description | Long Text Area | Event details | | | Volunteers Needed | Number | How many volunteers are required? |

| **Volunteer Assignment** | Volunteer (Lookup) | Lookup (Volunteer) | Links to the volunteer | | | Event (Lookup) | Lookup (Event) | Links to the event | | | Role | Picklist | Role assigned to the volunteer | | | Status | Picklist | Options: Assigned, Confirmed, Cancelled | | | Hours Worked | Number | Total hours contributed |

**✅ Step 3: Configured Page Layouts & Related Lists**

* Made sure that **Volunteer Assignments** appear in the **Related Lists** of:
  + **Volunteer Object** (to track their event assignments)
  + **Event Object** (to track assigned volunteers)

**✅ Step 4: Built the Flow for Auto-Updating Volunteer Status**

* **Goal:** Automatically update the **Volunteer’s Status to "Active"** when they are assigned to an event.
* **Created a Record-Triggered Flow**:
  + **Object:** **Volunteer Assignment**
  + **Trigger:** **When a record is created**
  + **Action:** Find the related **Volunteer** and update their **Status to "Active"**.

**❌ Key Limitation We Identified**

* **The Flow only triggers when a new Volunteer Assignment is created.**
* **If you edit an existing Volunteer Assignment, the Flow does NOT run.**
* **To fix this, we updated the trigger to "Record Created or Updated" so it works for both new & existing assignments.**

**✅ Step 5: Fixed the Flow to Handle Existing Volunteers**

* **Updated the Flow trigger to:** "A record is created or updated".
* **Now, if a new assignment is created OR an old one is edited, the Flow runs.**
* **This ensures that existing Volunteers also get their Status updated.**

**📌 Final Project Status**

✅ **Volunteer Management App is fully functional.**  
✅ **Volunteer Assignments automatically update the Volunteer Status.**  
✅ **Works for both new and existing Volunteer Assignments.**  
✅ **Now ready for Portfolio Deployment.**