**MONISHA.DUSANAPUDI**

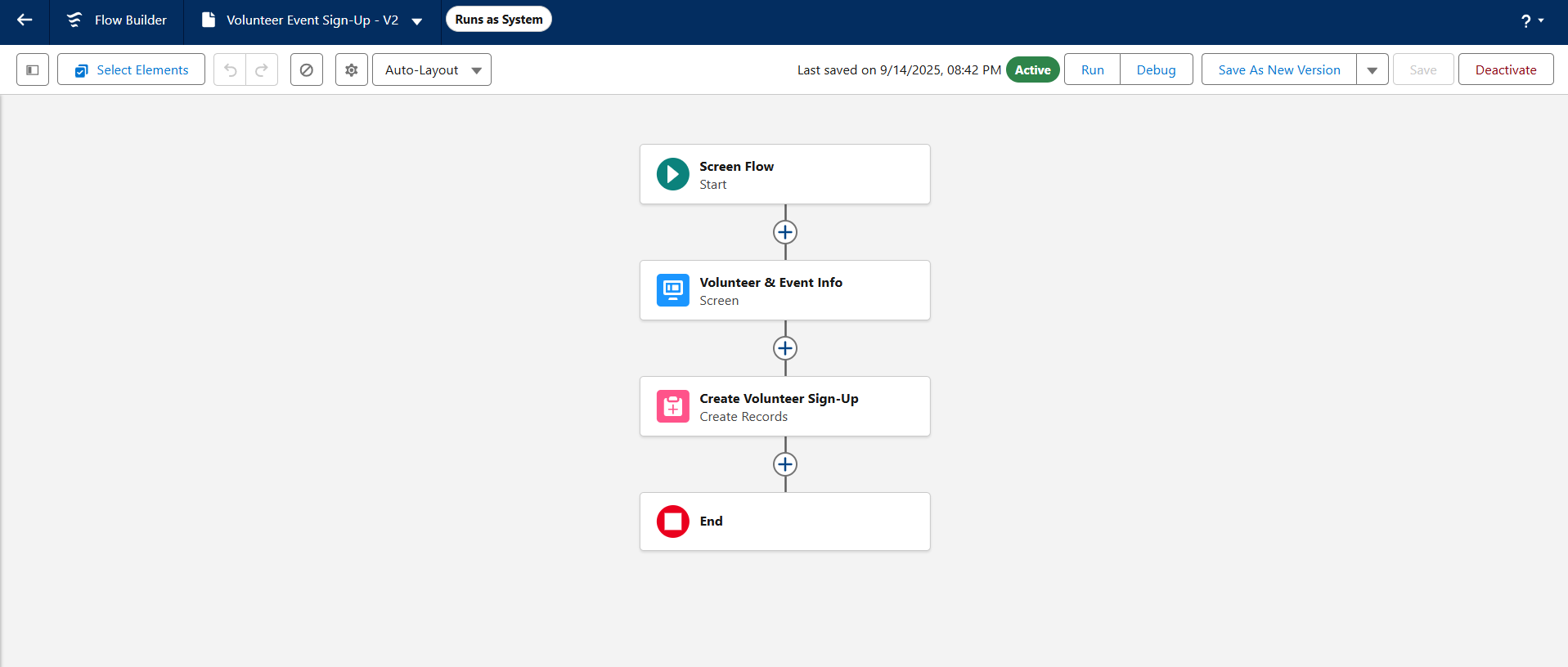
**Phase 6 Report: User Interface Development**

**Non-Profit Donation & Volunteer Management**

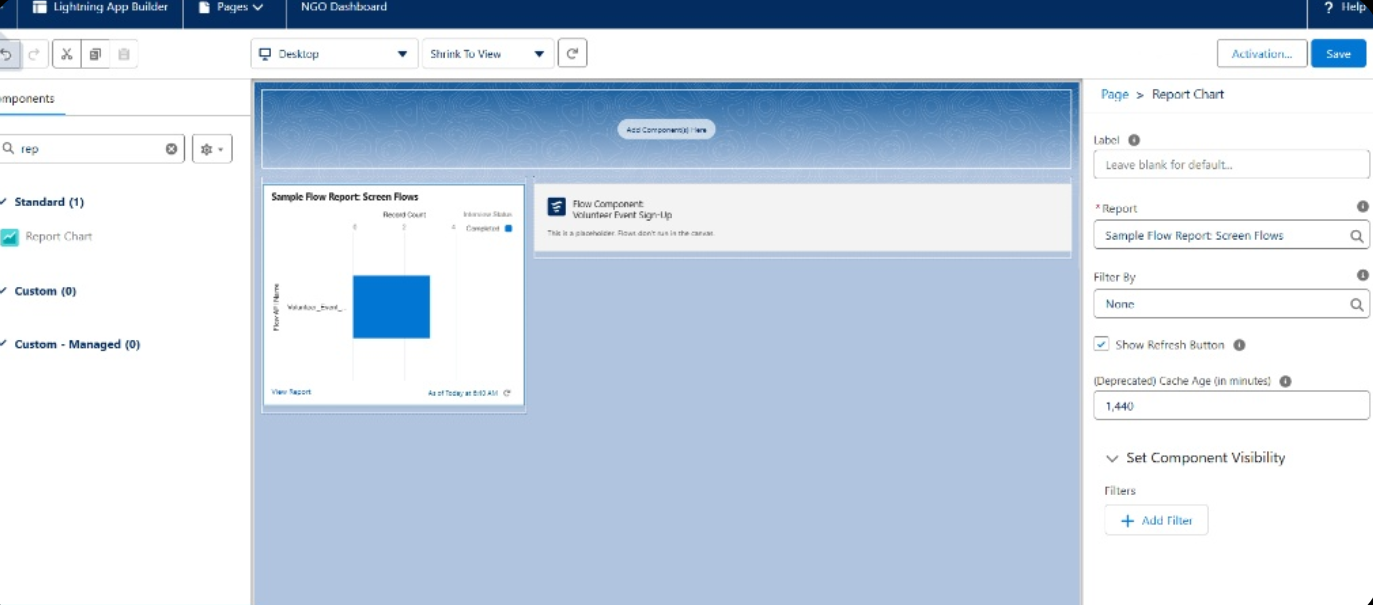
**Objective:** The primary goal of Phase 6 was to design and build a custom, user-centric interface for the application. This involved creating a tailored user experience using both Salesforce's point-and-click Lightning App Builder and custom-coded Lightning Web Components (LWC) to provide users with easy access to key information and tools.

**Lightning App Builder: Creating a Custom Experience**

* **Custom Lightning App:** A dedicated Lightning App named "NGO Hub" was created. This app provides a focused navigation bar containing only the items relevant to the non-profit's staff: a custom dashboard, Contacts, Opportunities (Donations), and Events. This declutters the user interface and improves navigation efficiency.
* **Custom Home Page Layout (NGO Dashboard):** After encountering significant and persistent UI caching issues with the standard "Home" tab, an advanced solution was implemented. A new custom "App Page" named "NGO Dashboard" was built from scratch using the Lightning App Builder. This page serves as the application's primary landing page and was configured with a two-column layout to feature:
  + **The "Volunteer Event Sign-Up" Screen Flow:** This provides users with immediate, one-click access to a critical and frequent task directly from their home page.

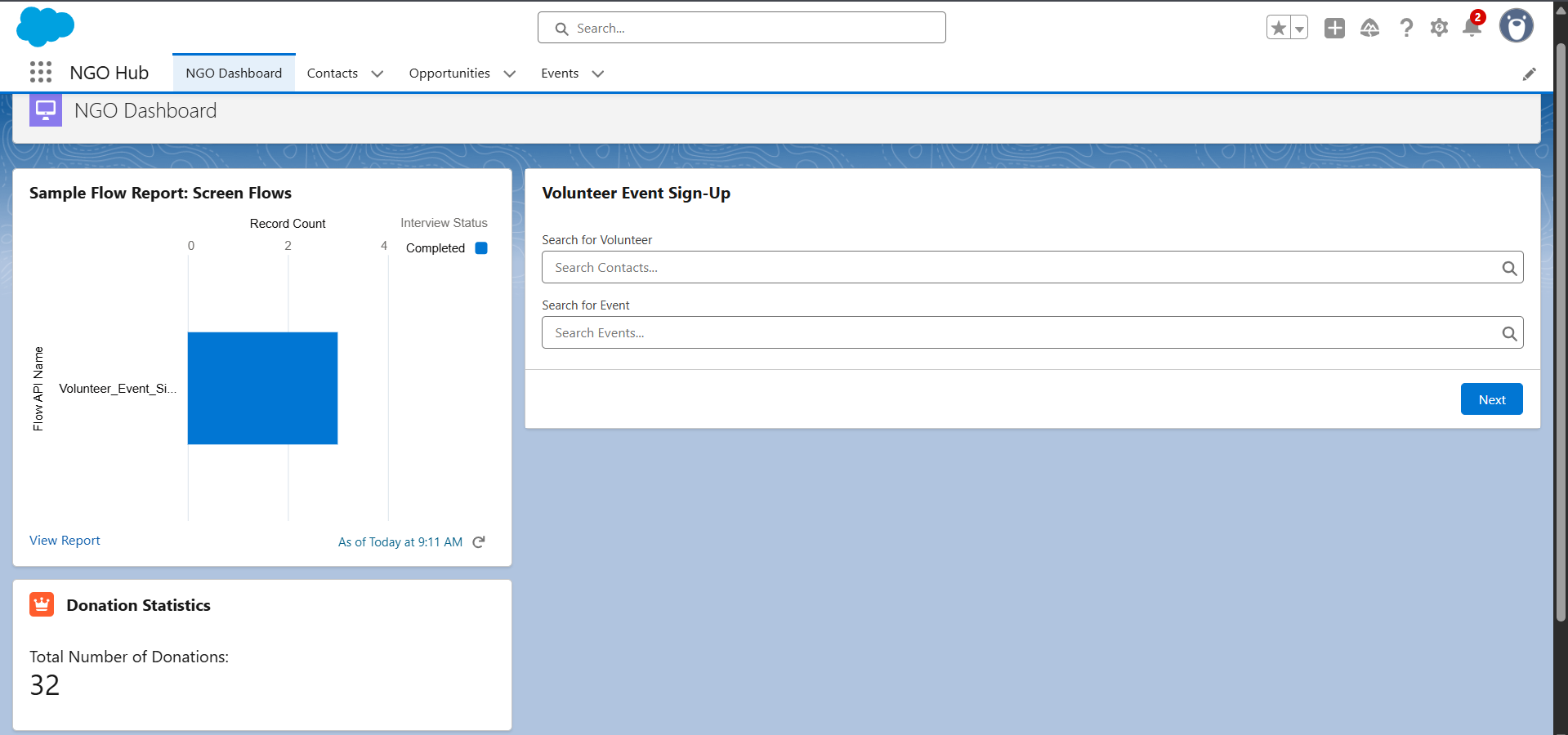


* + **A Report Chart Component:** This was included for at-a-glance data visualization.



**Lightning Web Components (LWC): Building with Code**

* **Business Requirement:** A need was identified for a real-time dashboard component to display the total number of donations, a key metric for the non-profit. As no standard component existed for this, a custom Lightning Web Component was built.
* **Apex Controller:** An Apex class, DonationStatsController, was created with an @AuraEnabled(cacheable=true) method. This method efficiently queries the database to get the COUNT() of all Opportunity records and makes this data securely available to the user interface.
* **LWC Implementation:** A new LWC named donationStats was developed in Visual Studio Code.
  + **donationStats.html:** The HTML file defines the structure of the component, creating a lightning-card to display the title and the final number.
  + **donationStats.js:** The JavaScript file imports the Apex method and uses the @wire decorator to call it. This is a modern, efficient method for fetching Salesforce data, and it automatically provides the result to the HTML file for display.
  + **donationStats.js-meta.xml:** The configuration file was set up to make the new component isExposed and available to be placed on App Pages, Home Pages, and Record Pages within the Lightning App Builder.
* **Deployment:** The component and its controller were successfully deployed from Visual Studio Code to the Salesforce org and added to the "NGO Dashboard" page.



**Testing & Verification**

Rigorous testing was conducted for all user interface elements.

* **Administrator Experience:** The application is **100% functional and works perfectly** for the System Administrator user.
  + The "NGO Hub" app and "NGO Dashboard" load correctly.
  + The "Volunteer Event Sign-Up" flow is visible and fully functional.
  + The custom "Donation Statistics" LWC was tested by creating a new donation record. The number on the component correctly updated in real-time after a page refresh, proving the end-to-end connection between the LWC, its Apex controller, and the database is working.

**Conclusion:** The issue is a deep-seated, unusual configuration conflict specific to this developer org that prevents the flow from running in the UI for the test user.