



STRATEGIC ICT & EBUSINESS IMPLEMENTATION: MSc in DA

**PART II : PROJECT IMPLEMENTATION REPORT
20TH APRIL 2019**

TEAM MEMBERS:

STUDENT NAME	STUDENT NUMBER	WORKLOAD DISTRIBUTION
ADITYA ARORA	X18132120	25%
HARSH JANGRA	X18139248	25%
JAYDEEP DEKA	X18134246	25%
KUNAL JANGID	X18129005	25%

SUBMITTED TO: CIARAN HAYDEN

Contents

Business Background:	4
The Problem Statement:	4
Current Available solutions:	6
Project Scope:	6
Limitations:	7
Future Scope:	7
Infrastructure:	8
Business Requirement:	8
Process Flow:	8
Database Design	9
Schema Diagram	9
Donor Table	10
Shelter Home Table	10
Request Table	10
Pledges Table	11
Process Work Flow charts	12
User Registration Flow	12
User Login Flow	13
Request Creation Flow	14
Pledge Creation Flow	15
Implementation	16
Application Flow	17
Custom Objects	17
Welcome Page	17
Donor Account creation	18
Shelter Home Account creation	18
Email Id Validation rules	19
Password Validation rules	19
Confirmation mail on successful account creation	20
Login Page	21
Login Validation	21
Donor Account Page and Pledge creation	22
Apex code for pledge creation	22
Shelter Home Account page and Request creation	24
Apex code for Request creation	25

Admin View for accepting and rejecting a shelter home.....	27
Approval mail for shelter home	27
Approval Mail process builder	28
Updated pending list for shelter homes	28
Test Data:.....	29
Reporting and Dashboards	32
Operational Reports	32
Management Dashboards.....	34
Dashboard 1: Overview.....	34
Dashboard 2: Pledge Overview.....	35
Dashboard 3: Request Overview	36
References	38
Appendix.....	39

Business Background:

'Make it Count' is an Integrated Donations Management Platform, which brings together Shelter Homes and their donors to a single platform.

The application makes it easier for shelter homes to access a large base of donors, and provides the donors with a convenient way to identify the requirements at social establishments around them.

The application allows shelter homes and donors to create requests or pledges, respectively, and then view these entries and respond accordingly.

This provides an effective medium to match the requirements at the shelter homes to availability with the donors in the community.

We essentially aim to bring a market-place based model to donations management at shelter homes.

The Problem Statement:

Lack of affordable housing, unemployment, poverty, and other socio-economic pressures have been causing a rise in homelessness across cities. There are over 9,000 homeless individuals in Dublin alone, as of June 2018. (<https://www.thejournal.ie/homeless-ireland-4-4097015-Jun2018/>).

There are several organizations working to help these individuals. Apart from government funding, these organizations rely heavily on donations from the general public and business establishments, to provide the best possible help to the homeless population.

Due to unavailability of a platform to communicate their immediate needs to the public, donations either reach too late, or remain underutilized elsewhere.

Moreover, people willing to make donations may find it difficult to find an organization with matching requirements. It may not be uncommon, for example, for an organization to receive food when they were actually out of appropriate storage space, but did need some additional blankets at that point.

As seen during the 2008 Kerala floods in India, thousands of people made their contributions to the relief effort. While a few centres received excess donations, others still struggled to fulfil their requirements.



We have seen in past that many a times needy haven't been looked after.



Donor retention is an important metric for any not-for-profit organization. Identifying regular donors, following up with new donations, etc are important analytical tasks that will help the organizations function better. Such requirements make for a strong use-case for the organizations to use a CRM-like application.

Donor retentions can be increased:

- By offering many new ways and categories to donate.
- By always opting P2P donation campaigns.
- By following up with the lapsed donors.
- By maintaining continuous communication.

Our main aim is to build an application which can be accessed through web by both donors and shelter homes, and bridge that gap between both. The system will work closely together with different organisations and individuals to provide near-real time update to both the parties. We aim to optimize the flow of aid from donors to shelter houses such that we can minimize wastage and ensure that pledged items reach where they are being optimally utilized.

The application also provides analytical capabilities to generate report on past donations.

Current Available solutions:

At present there are few options available in the market which provides “CRM services” and help in making “Donations”. But as their names suggests they are divided and categorized in two parts. Present top Donation Management Software and Application are listed below,

CRM for Donation Management (Donation Management Software)-

- Bloomerang.
- Raiser's Edge NXT.
- DonorPerfect.
- Network for Good.
- eTapestry.

Applications For making donation-

- Givi Lets Donors Easily Search for Nonprofits.
- One Today Highlights New Nonprofits All the Time.
- Givelify Makes Donating to Nonprofits Smooth.
- uBack Connects Donors to Causes They Care About.
- Coin Up Gathers Micro Donations to Make Change.

These solutions and applications help in managing relationship among Donors and recipients, which include managing donor information and their transactions.

Project Scope:

We have identified several gaps with the way shelter homes collect donations:

1. Difficulty in communicating immediate requirements to the society
2. Time taken to request, and receive donations can be too high
3. Donations may not reach where they are required the most immediately
4. Donors may find it difficult to find an organization that could use their help
5. Lack of a common platform to enable engagement between the shelter homes and the society

The application aims to address each of these issues systematically. Specific functionalities from within the Salesforce CRM have been utilized to build a system that caters to the requirements from both, donor and shelter home perspective.

The application needs to adhere to the following project goals:

1. Users are able to log in as donors / shelters home using web integration. Donors can register as either individuals or organizations
2. Users are able to create pledges / requests for a specific category using web integration.
3. Users are able to create pledges / requests in response to specific requests / pledges using web integration
4. Allow creation of campaigns to facilitate donations for a specific purpose (for ex, preparation for upcoming storm warning, etc.)
5. System generated reports help match pledges / requests based on selected categories, and their status is updated as such
6. Analytical reports on activities on the platform
7. Scheduled report provides frequent updated on open requests / pledges.

Limitations:

- In the scenario where users do not provide an existing pledge / request name, the matching needs to be done by the administrator. This causes a manual intervention, and may cause delays in the matching process
- Reports from the application provide data on requests and pledges that are currently open. This information needs to be hosted and updated online separately. There is no convenient way to integrate this into the application

Future Scope:

Several functionalities can be added to the application to enhance user experience, and achieve greater integration:

- Integration with Payment Gateway to enable cash donations
- Web API to integrate with business websites, helping create corporate partnership
- Mobile application
- Automated alerts on matching pledges and alerts
- Greater integration with the transportation process, once pledges and requests are matched
- Categorize Donor organizations based on business type for focused engagement
- Implement Salesforce Communities to promote greater engagement
- Access to Blackbaud community
- Include a wider range of non-profit organizations

Infrastructure:

- Considering the upfront cost of hardware, it is better to go for cloud-based service than on-premise. Cloud-based services are scalable as per requirement.
- Software as a Service is preferred over Platform as a Service as the former needs licensing cost, time to time software update.
- A cloud-based Customer Relationship Management (CRM) service will be more beneficial as it reduces development and deployment time.
- Salesforce applications have feature of API integration that can be used later in other application.
- The Analytics service to create dashboards will help to get insights of different transactions

Business Requirement:

The below points describe the overall business objective the application aims to fulfil:

- Shelter Homes and Donors can register on a common platform
- Pledges and Requests can be created via web integration by donors and shelter homes respectively
- Ability to create campaigns to aggregate donations and run separate donation programs for donor or shelter home
- Pledges and Requests can be viewed and responded to. For example, donors can create pledges to fulfil a particular existing requirement
- Reports providing regular updates on open pledges / requests
- Perform analysis on platform usage, and identify major trends (top donors, etc.)

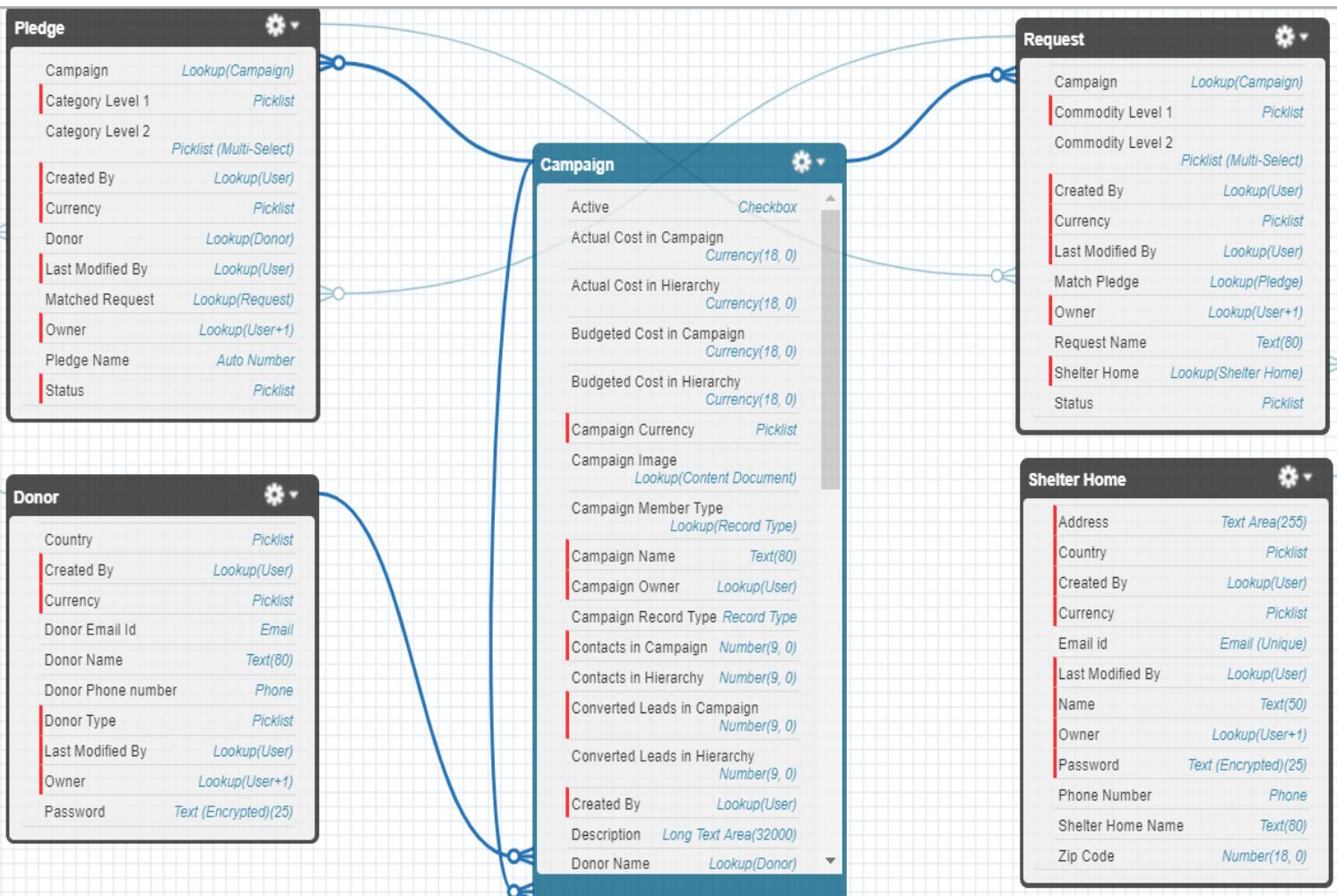
Process Flow:

- User enters the ‘Welcome Page’, and chooses Member Type as ‘Donor’ or ‘Shelter Home’
- User redirected to log in page for Donor or Shelter Home as per selection in the first step. New users can navigate to Registration Page and complete registration. Donors can register either as an organization or an individual
- Donors are immediately notified on email post account registration. Shelter Homes are verified for authenticity, until then, the Shelter Home account status is ‘Pending’
- Admin verifies the Shelter Home registration and updates status to ‘Approved’. Shelter Home contact is notified via email, and can now start using the account
- User creates new Pledge / Request. A Request / Pledge name can be provided on the Pledge / Request, respectively, using a Lookup field provided in the creation form
- New request / pledge is created, and status is updated as ‘Matched’, if Pledge Name / Request Name is provided by user

- New request / pledge is created, and status is updated as 'Open', if Pledge Name / Request Name is not provided by user
- Matches with new pledge / request are identified based on reports generated in the application
- Status of matched pledges / requests is updated by system admin with Salesforce access. The corresponding Donor / Shelter Home is notified of the change in status of their pledge / request
- Report is generated with latest updated data on open requests / pledges
- Use Salesforce Reporting functionalities to identify Top Donors by number of pledges, Top shelters by number of requests, Top categories for which requests are created to enable targeted campaigns, Shelter Homes with the high amount of open requests, to enable personalized assistance if required. These reports are generated and described in the 'Reporting and Dashboards' section

Database Design

Schema Diagram



Donor Table

Sr.no	Column Name	Field Type	Description
1	Donor_type	Text	Classification of Donor (As we have 3 types)
2	Name	Text	Donor's Name
3	Email_id	Email	Donor's Email for registration
4	Password	Text (Encrypted)	Donor's Account Password
5	Phone_no	Phone	Donor's phone no
6	Country	Picklist	Donor's Address segments

Shelter Home Table

Sr. no	Column Name	Field Type	Description
1	Shelterhome_name	Text	Shelter Home's Name
2	Name	Text	Contact name
3	Email_id	Email	Shelter Home's Email
4	Password	Text (Encrypted)	Shelter Home's Account password
5	Address	Text Area	Shelter Home's Address
6	Phone no	Phone	Shelter Home's Phone no.
7	Zip code	Number	Shelter Home's Address's segments
8	Country	Picklist	Donor's Address's segments
9	Status	Picklist	Approval Status for Shelter Home

Request Table

Sr. no	Column Name	Field Type	Description
1	Commodity Level 1	Picklist	Main type of the commodity
2	Commodity Level 2	Multi-picklist	Sub Category of the commodity
3	Match Pledge	Lookup Relationship	Pledge Name
4	Request Name	Text	Request Name
5	Shelter Home	Text	Shelter Home name
6	Status	Picklist	Request Status

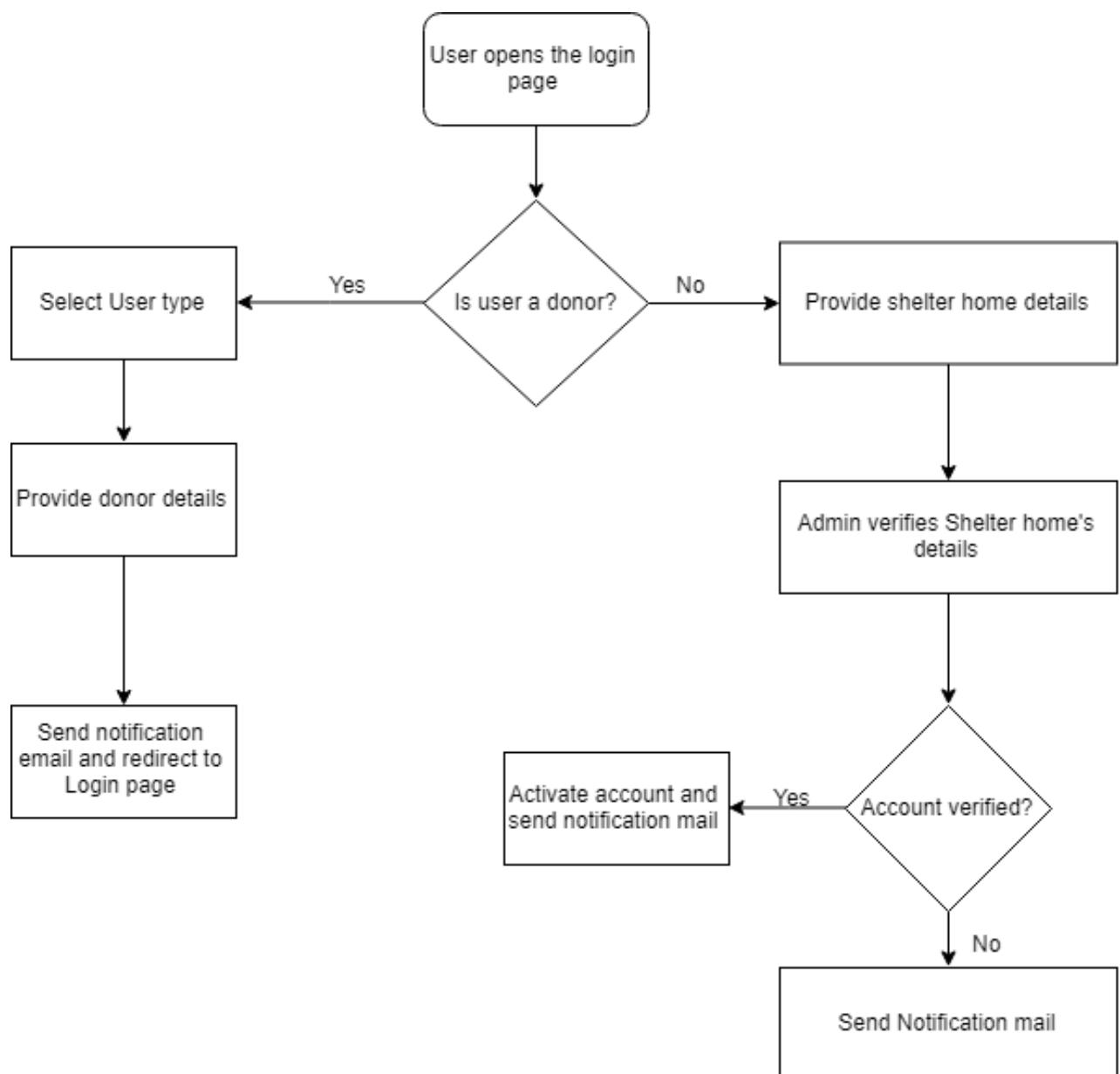
7	Campaign	Lookup Relationship	Campaign Name
----------	----------	---------------------	---------------

Pledges Table

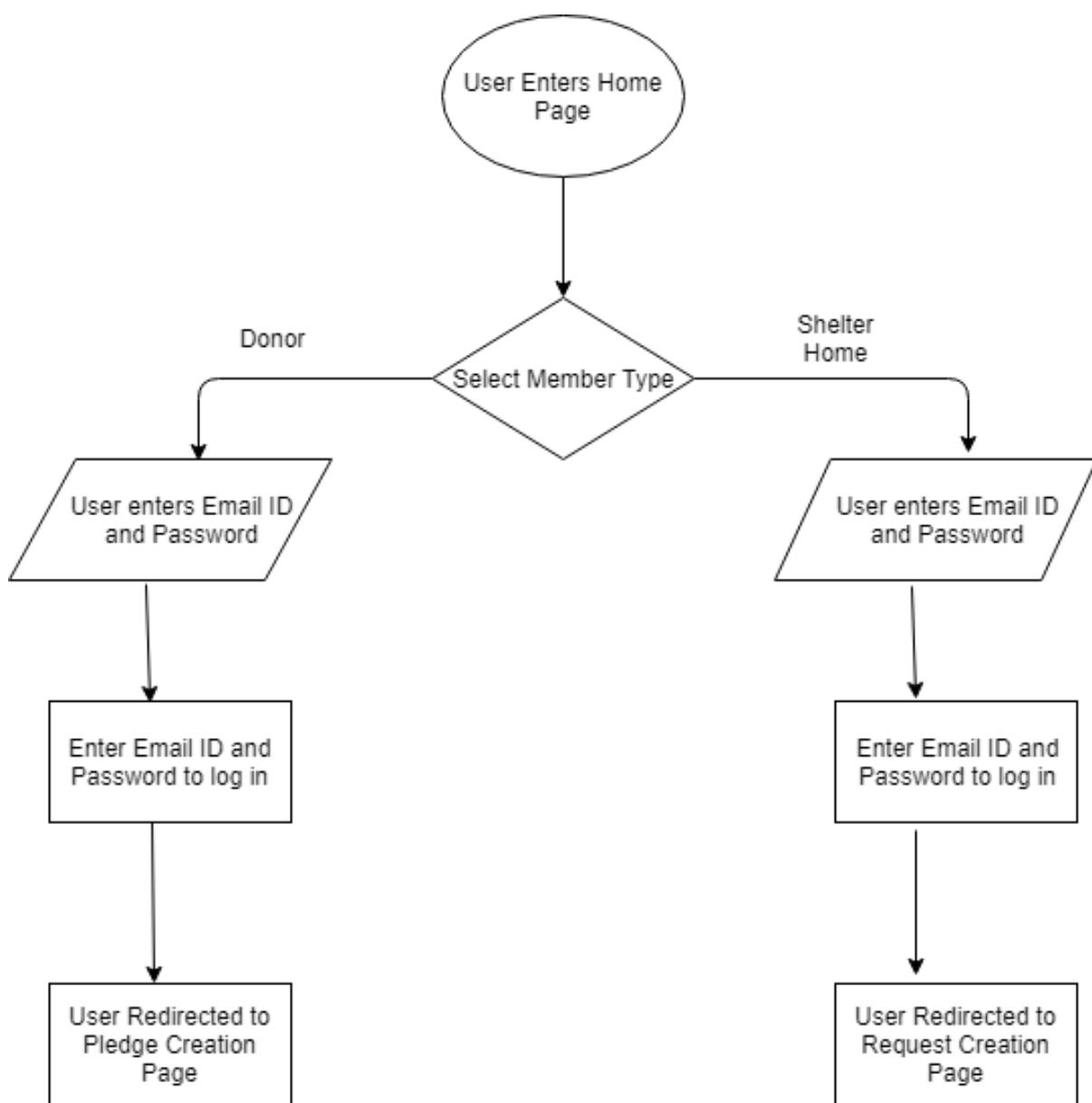
Sr. no	Column Name	Field Type	Description
1	Category Level 1	Picklist	Main type of Category
2	Category Level 2	Multi-picklist	Sub Category
3	Match Request	Lookup Relationship	Request Name
4	Pledge Name	Text	Pledge name created
5	Status	Picklist	Pledge Status
6	Donor Name	Lookup Relationship	Donor Name
7	Campaign	Lookup Relationship	Campaign Name

Process Work Flow charts

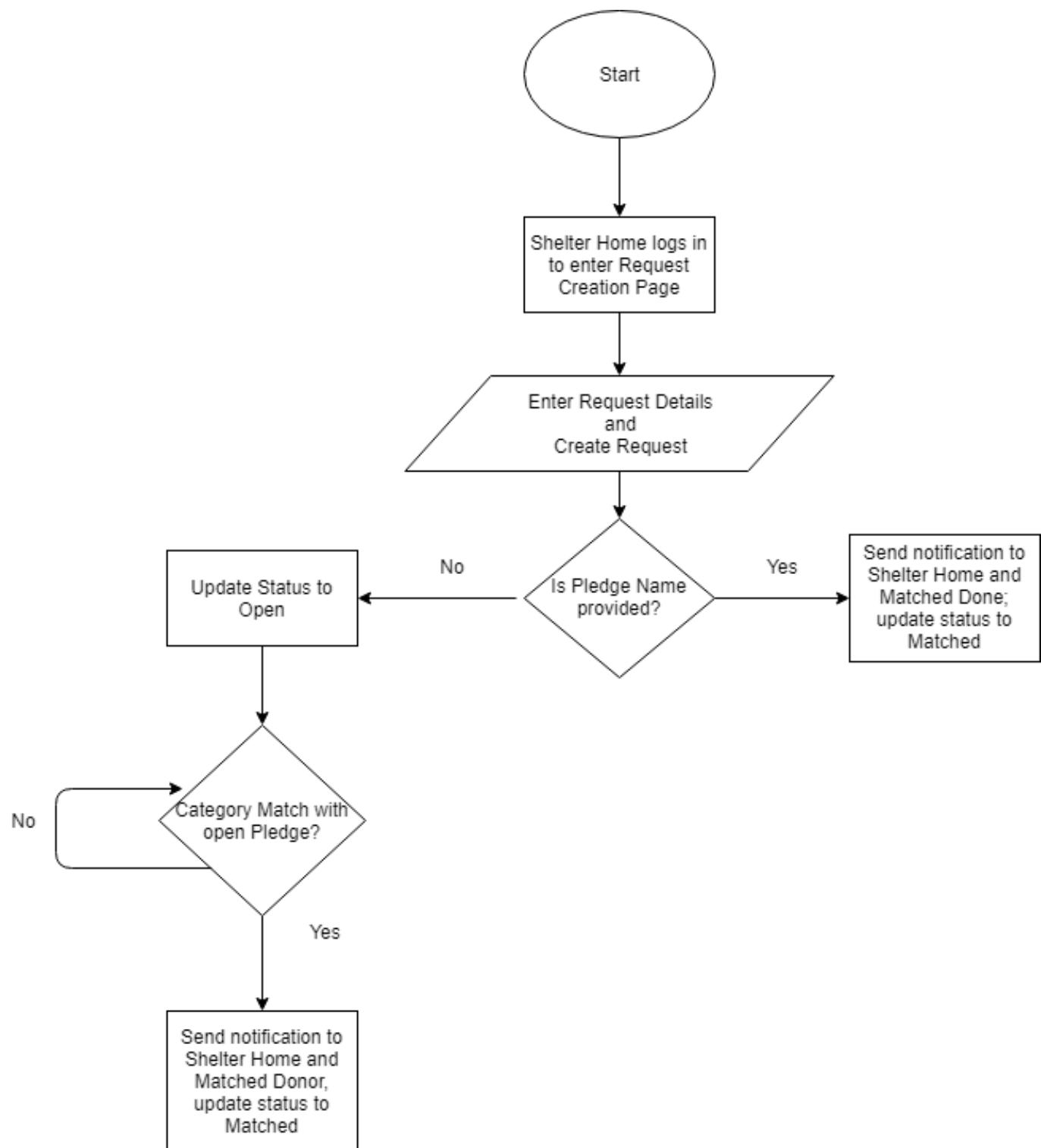
User Registration Flow



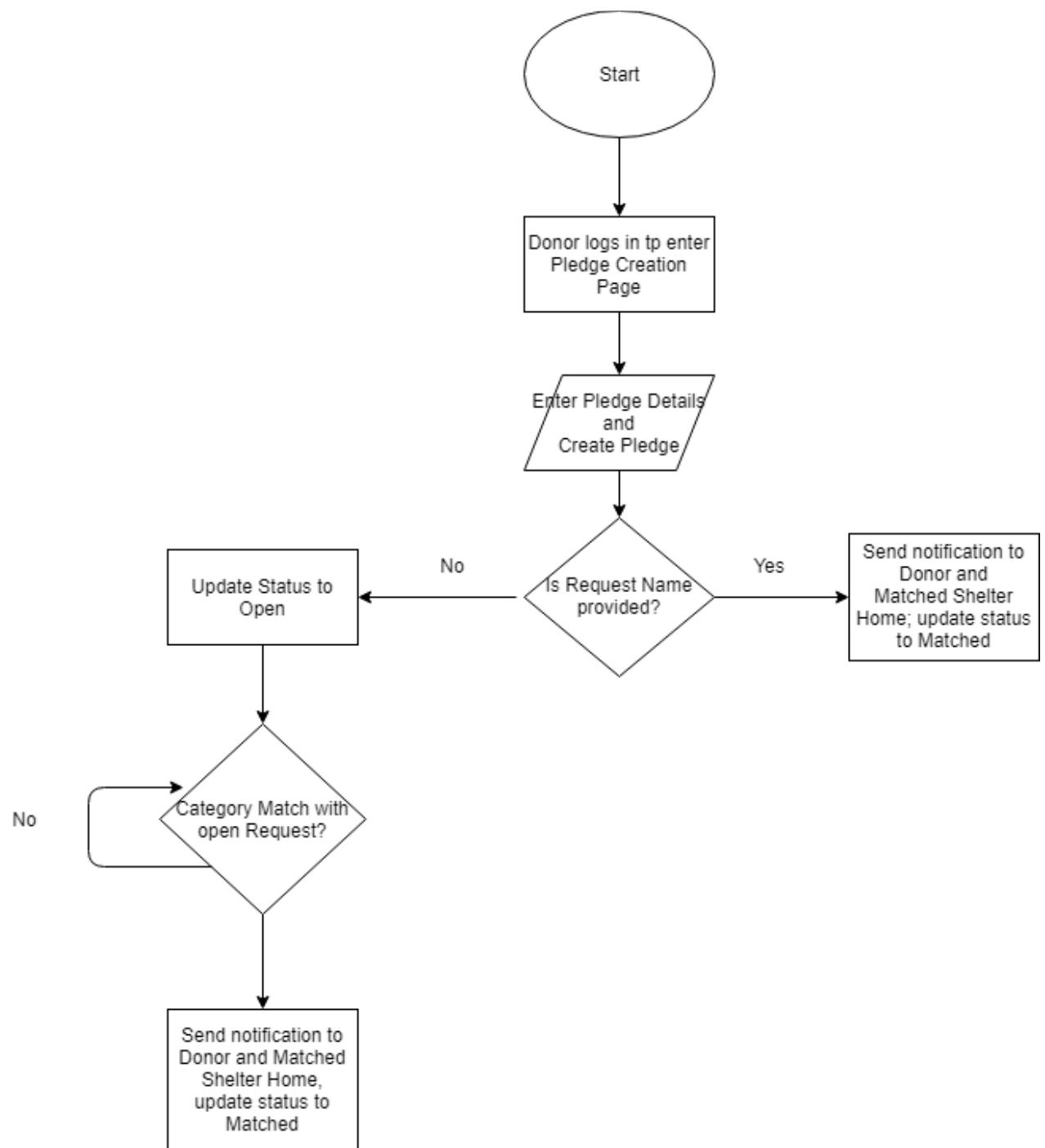
User Login Flow



Request Creation Flow



Pledge Creation Flow



Implementation

The project is implemented on Salesforce CRM software. Various functionalities within the system are utilized to implement the required features of the application. Visual Force pages are used to provide web integration.

The below points summarize the Implementation

- Custom objects are created for Donors, Shelter Homes, Pledges and Requests.
- Field dependencies are set to create the correct relations in the schema.
- Visual Force pages are used to provide Web Integration for User Registration, User Login, and Pledge / Request Creation. Custom controllers are used to send notification emails, and redirect to the next page.
- Salesforce validations are used to ensure users provide a valid email ID, and password as per requirement, during user registration.
- Email triggers are utilized to notify users of account creation / account approval (in case of shelter homes).
- Pledge Name can be selected while creating a Request, and vice-versa.
- ‘Campaign’ Standard field is utilized to create campaigns for donations.
- Campaigns use Record Type as ‘Donation’ or ‘Fundraiser’ to provide the functionality from the perspective of both donors and shelter homes
- Pledges and Requests can be tagged to a pre-existing campaign during creation, using a lookup field on the creation form.
- Status Flag is used to identify current status of Pledge / Request.
- Administrator account updates status from ‘Open’ to ‘Matched’ on identifying a match. History Logging is enabled on this field to ensure manual changes are tracked and can be accessed if required.
- Process Builder is utilized to trigger notification email on change in ‘Status’
- Salesforce analytics and reporting capabilities are utilized to create reports providing insights into user activity on the platform.
- Schedule Report functionality is used to provide latest data on open requests / pledges.

Application Flow

Custom Objects

Four different custom objects are created to store information about Donor, Shelter Home, Pledge and Request as shown below:-

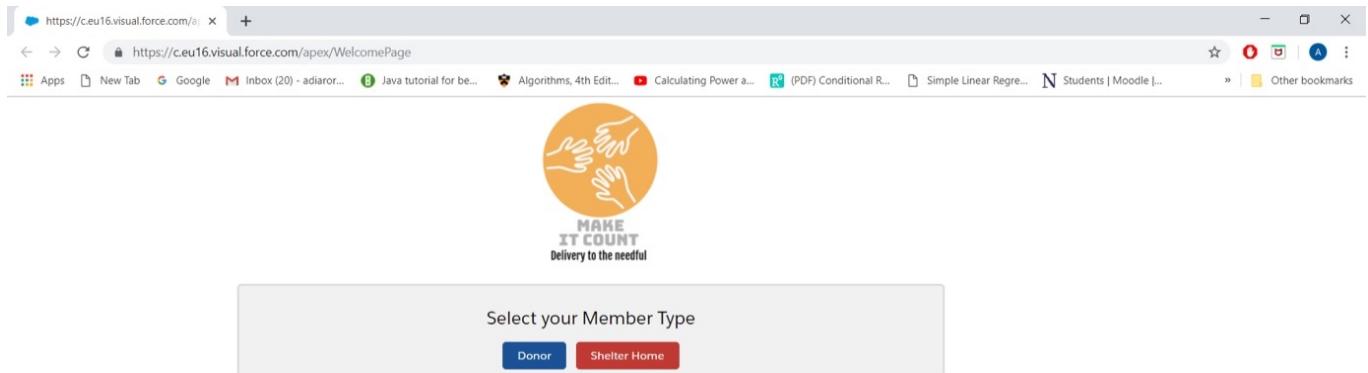
Custom Objects

Custom objects are database tables that allow you to store data specific to your organization in Salesforce. You can use custom objects to extend Salesforce functionality or to build new application functionality. Once you have created a custom object, you can create a custom tab, custom related lists, reports, and dashboards for users to interact with the custom object data. You can also access custom object data through the API.

Action	Label	Master Object	Deployed	Description
Edit Del	Donor		✓	
Edit Del	Pledge		✓	
Edit Del	Request		✓	
Edit Del	Shelter Home		✓	

Welcome Page

User selects either Donor or Shelter Home option from this page for Login or Registration.



Donor Account creation

Donor fill their details for account creation and while creating this account they need to select their type i.e. either Individual or organization as shown below:-

The screenshot shows a web browser window with a URL starting with <https://ceu16.visual.force.com/apex/RegistrationPage?sfid.tabName=01r1t000000AV15>. The main content is a form titled "DONOR REG FORM". It contains fields for "Full Name", "Phone", "Email", "Password", and a dropdown menu for "Donor Type" with options like "None". At the bottom are "Create Account" and "Cancel" buttons.

After clicking on Create Account, Donor receive a confirmation email for successful account creation.

Shelter Home Account creation

Shelter Home fill their details for account creation on the page shown below:-

The screenshot shows a web browser window with a URL starting with <https://ceu16.visual.force.com/apex/ShelterHomeReg>. The main content is a form titled "Shelter Home Registration Form". It contains fields for "Full Name", "Phone", "Email", "Password", "Address", "Country", and "Zip Code". At the bottom are "Create Account" and "Cancel" buttons.

And after clicking on Create Account button, the Shelter Home wait for account confirmation email, post verification by administrator.

Email Id Validation rules

Following rules are tested at the back end for email validation:-

Rule 1: Email Id should be a valid one.

Rule 2: Email should not be registered earlier.

The screenshot shows a registration form titled "DONOR REG FORM". The fields are as follows:

- Full Name: XYZ Company
- Phone: Phone No
- Email: abc (highlighted in red)
- Error message: Error: Invalid Email Address.
- Password: Password
- Donor Type: Organization, ▾

At the bottom are two buttons: "Create Account" (blue) and "Cancel" (red).

As we can see, while entering an invalid email id, the system prompts an error by giving error message “Invalid Email Address” and it behaves in similar manner if an already used email id is used by a user while registration.

Password Validation rules

Following rules are tested at the back end for password validation:-

Rule 1: Password length should be of minimum six character.

Rule 2: Password should contain at least one character and one digit.

As we can see, while entering an invalid password on the basis of two rules stated above, the system prompts an error by giving error message “The password should contain at least one digit and one character; minimum length 6 characters”.

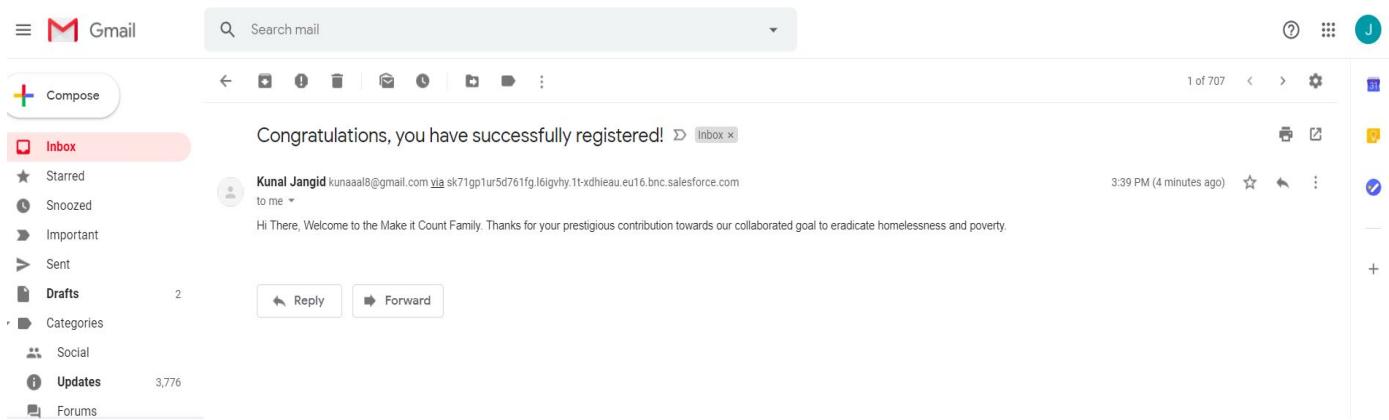
The screenshot shows a Salesforce interface for 'Donor Management'. A user is attempting to create a new donor with the following details:

- Donor Name: My Name
- Donor Type: Individual
- Donor Email Id: myname@email.com
- Password: sdf4/ (which is invalid)
- Donor Phone number: (empty)
- Country: --None--
- Currency: EUR - Euro

An error message is displayed: "Error: Invalid Data. The Password should contain at least one digit and one character; minimum length 6 characters".

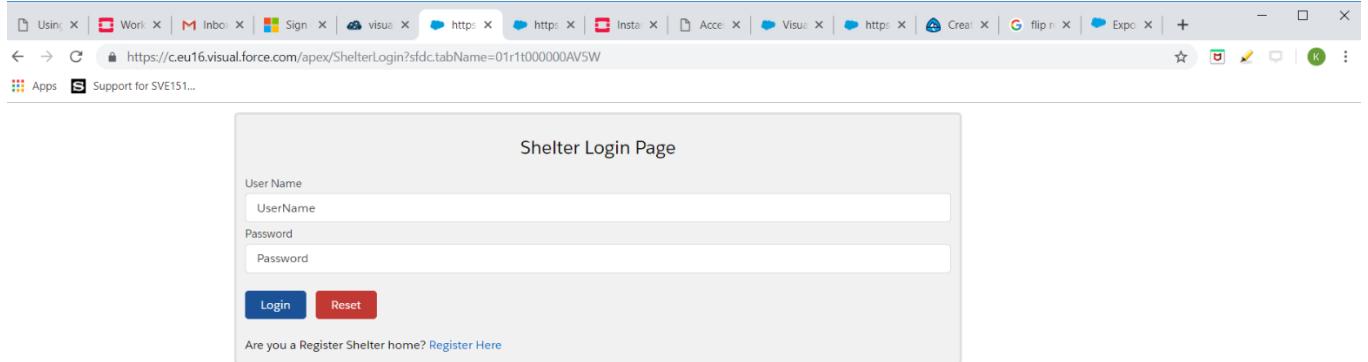
Confirmation mail on successful account creation

Once a user submit their valid details on the Registration page, a confirmation mail is triggered to their registered email id as shown below,



Login Page

For logging in, user enter their credentials on Login page. Donor and Shelter home both have separate login pages. The Login page shown below is for Shelter Login Page.



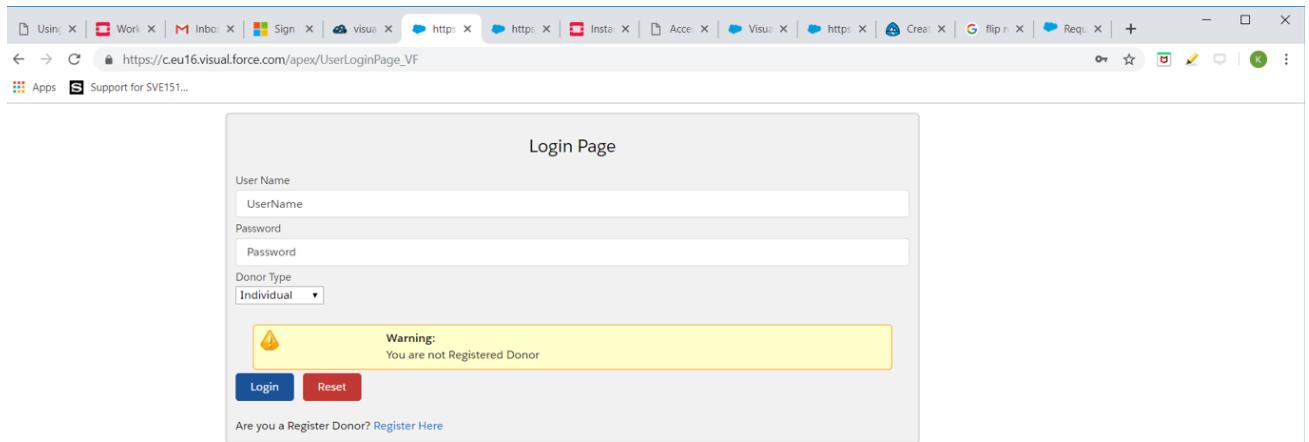
The screenshot shows a web browser window with the URL <https://ceu16.visual.force.com/apex/ShelterLogin?sfid.tabName=01r1t000000AV5VW>. The title bar says "Shelter Login Page". The page contains two input fields: "User Name" (labeled "UserName") and "Password". Below the password field is a dropdown menu labeled "Donor Type" with "Individual" selected. At the bottom are "Login" and "Reset" buttons, and a link "Are you a Register Shelter home? [Register Here](#)".

Login Validation

While entering the login credentials by a user, if the credentials are valid, user is able to login to their account.

But if the credentials are wrong or invalid, user gets an error message stating “You are not a registered donor”.

Same has been shown in the below screenshot.



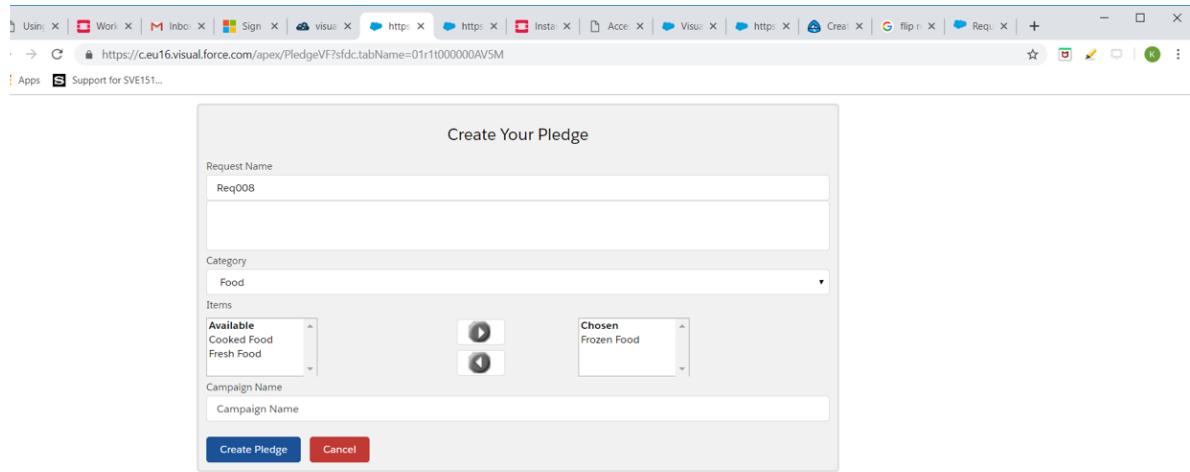
The screenshot shows a web browser window with the URL https://ceu16.visual.force.com/apex/UserLoginPage_VF. The title bar says "Login Page". The page contains three input fields: "User Name" (labeled "UserName"), "Password", and "Donor Type" (dropdown menu with "Individual" selected). Below the password field is a dropdown menu labeled "Donor Type" with "Individual" selected. A yellow warning box appears with the text "Warning: You are not Registered Donor". At the bottom are "Login" and "Reset" buttons, and a link "Are you a Register Donor? [Register Here](#)".

Donor Account Page and Pledge creation

After successful login to the account, the Donor can straight away create their pledges on “Create Your Pledge” page.

Also, the donor can view the Requests listed by each Shelter home. To do so, the donor need to select an existing request from “Request Name” Lookup Field and create a pledge to fulfil the selected Request.

Once a new match is identified for the pledge based on the provided Category, and administrator updates the Pledge Status.



Apex code for pledge creation

The above page is created using Visual Force pages.

This enables web integration, so that users are able to create a pledge without signing in to the Salesforce application.

Providing web integration to create pledges /requests is a core objective of the application, and these Visual Force pages are used to fulfil the same.

```
<apex:page standardStylesheets="false" showHeader="false"
standardController="Pledge__c" lightningStylesheets="true">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-----Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
    }
</style>
```

```

        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }

```

</style>

<!--Style Ends-->

<apex:form style="margin-left:30%;margin-right:-30%;">

<div class="container">

<div class="slds-card_header">

<p style="font-size:20px;text-align:center;">Create Your Pledge</p>

</div>

<div class="slds-form-element">

<label class="slds-form-element_label" for="input-id-01">Request Name</label>

<div class="slds-form-element_control">

<apex:inputField id="input-id-03" value="{!!Pledge__c.Matched_Request__c}" styleClass="slds-input" html-placeholder="Request Name"/>

</div>

</div>

<div class="slds-form-element">

<label class="slds-form-element_label" for="input-id-01">Category</label>

<div class="slds-form-element_control">

<apex:inputField id="input-id-01" value="{!!Pledge__c.Pledge_Category_Level_1__c}" styleClass="slds-input" html-placeholder="Category"/>

</div>

</div>

<div class="slds-form-element">

<label class="slds-form-element_label" for="input-id-01">Items</label>

<div class="slds-form-element_control">

<apex:inputField id="input-id-02" value="{!!Pledge__c.Category_Level_2__c}" styleClass="slds-input" html-placeholder="Items"/>

</div>

</div>

```

<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">Campaign Name</label>
    <div class="slds-form-element__control">
        <apex:inputSecret lang="" id="input-id-04" value="{!!Pledge__c.Campaign__c}" styleClass="slds-input" html-placeholder="Campaign Name"/>
    </div>
</div>
<div>
    &nbsp;
</div>
<div>
    <apex:commandButton styleClass="slds-button slds-button_brand" value="Create Pledge" action="{!save}"/>
    &nbsp;
    <apex:commandButton styleClass="slds-button slds-button_destructive" value="Cancel" action="{!cancel}"/>
</div>

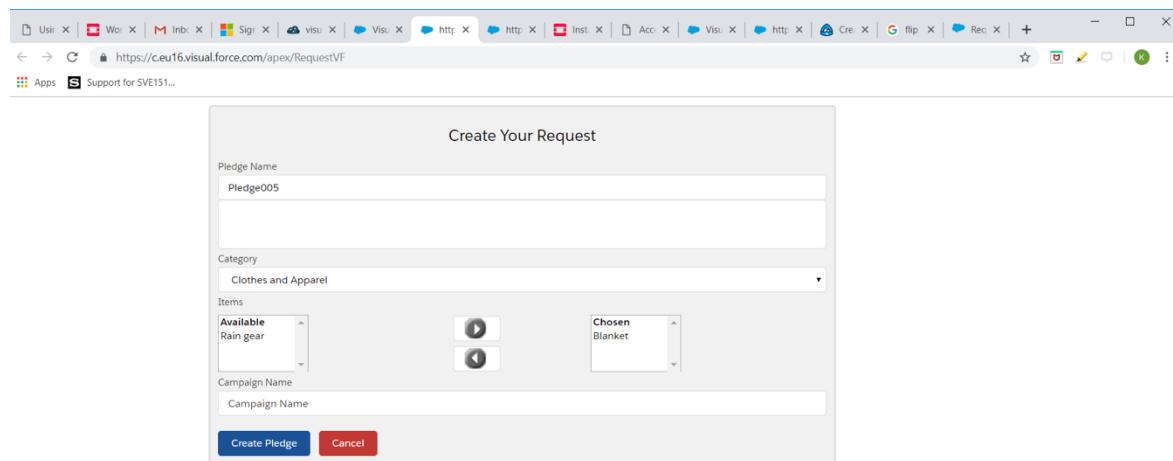
</div>
</apex:form>
</apex:page>

```

Shelter Home Account page and Request creation

After successful login to the account, the Shelter home has following options,

- Shelter home can select an existing donation (any available Pledge by a Donor) by selecting a “Pledge Name” from the dropdown.
- For creating a request, they can enlist their needs with the help of different “Categories” present in the dropdown option and the items listed in the “Item list”. A pre-existing Campaign Name can be tagged to the pledge, to aggregate donations made for a specific purpose, or under a specific program.



Apex code for Request creation

```
<apex:page standardStylesheets="false" showHeader="false"
standardController="Request__c" lightningStylesheets="true">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!----Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }
</style>
<!----Style Ends----->
<apex:form style="margin-left:30%;margin-right:-30%;">
<div class="container">
    <div class="slds-card_header">
        <p style="font-size:20px;text-align:center;">Create Your Request</p>
    </div>
    <div class="slds-form-element">
        <label class="slds-form-element_label" for="input-id-01">Pledge Name</label>
        <div class="slds-form-element_control">
```

```

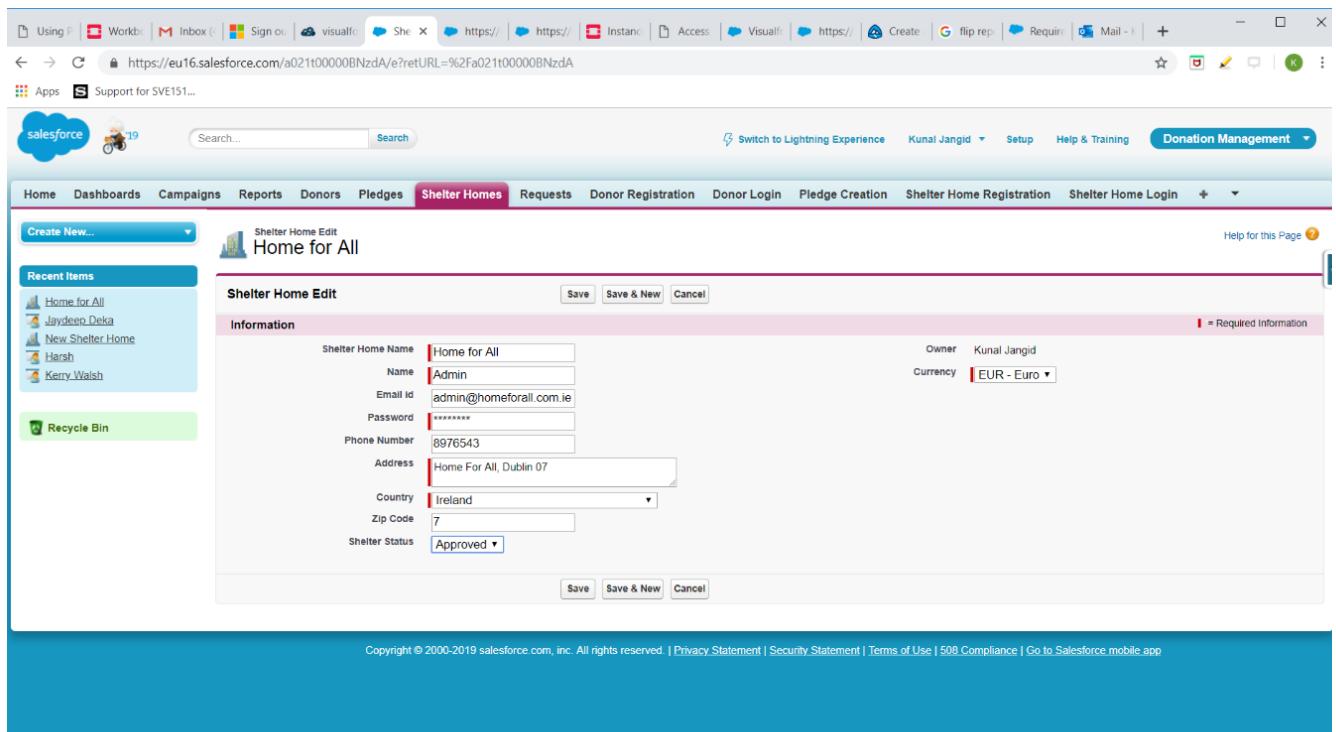
        <apex:inputField id="input-id-03" value="{!Request__c.Match_Pledge__c}"
styleClass="slds-input" html-placeholder="Request Name"/>
    </div>
</div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">Category</label>
    <div class="slds-form-element__control">
        <apex:inputField id="input-id-01"
value="{!!Request__c.Commodity_Level_1__c}" styleClass="slds-input" html-
placeholder="Category"/>
    </div>
</div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">Items</label>
    <div class="slds-form-element__control">
        <apex:inputField id="input-id-02"
value="{!!Request__c.Commodity_Level_2__c}" styleClass="slds-input" html-
placeholder="Items"/>
    </div>
</div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">Campaign Name</label>
    <div class="slds-form-element__control">
        <apex:inputSecret lang="" id="input-id-04" value="{!!Request__c.Campaign__c}"
styleClass="slds-input" html-placeholder="Campaign Name"/>
    </div>
</div>
<div>
    &nbsp;
</div>
<div>
    <apex:commandButton styleClass="slds-button slds-button_brand" value="Create
Pledge" action=" {!save} "/>
    &nbsp;
    <apex:commandButton styleClass="slds-button slds-button_destructive"
value="Cancel" action=" {!cancel} "/>
</div>

</div>
</apex:form>
</apex:page>

```

Admin View for accepting and rejecting a shelter home

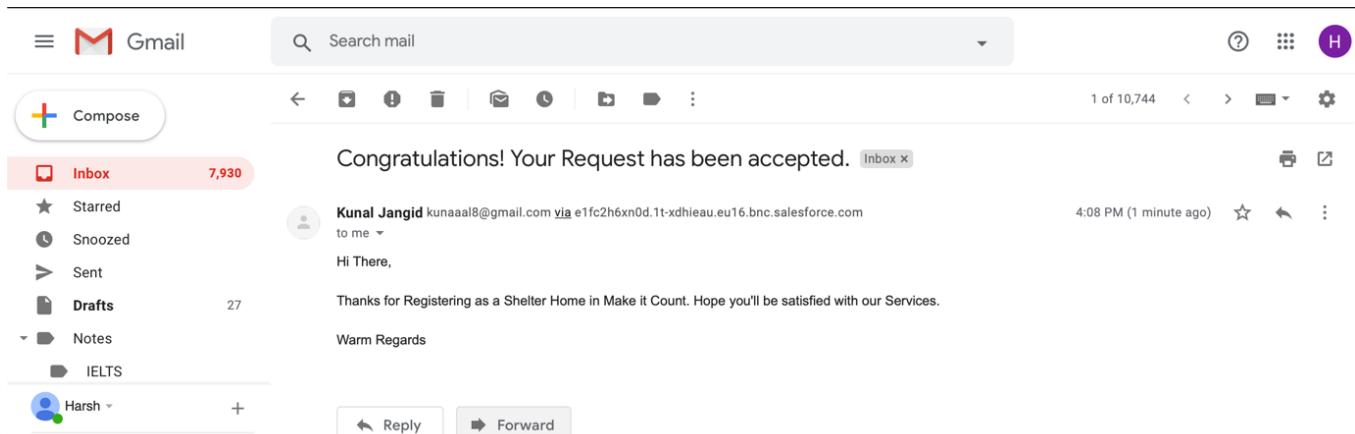
Shelter Homes with pending Verification are listed in the ‘Shelter Homes By Status’ report. The admin uses this report to verify the information for shelter homes with pending registration approval. Once, these are verified, the admin can choose the value “Approved” for the option “Shelter Status” as shown below,



The screenshot shows the 'Shelter Home Edit' page for 'Home for All'. The 'Shelter Home Name' is 'Home for All'. The 'Name' field contains 'Admin'. The 'Email Id' field contains 'admin@homeforall.com.ie'. The 'Phone Number' field contains '8976543'. The 'Address' field contains 'Home For All, Dublin 07'. The 'Country' field is set to 'Ireland'. The 'Zip Code' field contains '7'. The 'Shelter Status' dropdown is set to 'Approved'. The page also shows the owner as 'Kunal Jangid' and the currency as 'EUR - Euro'.

Approval mail for shelter home

Shelter home receives a triggered confirmation e-mail once they are approved by the admin after completion of verification process.



The screenshot shows an email in the Gmail inbox from 'Kunal Jangid' (kunaaal8@gmail.com) with the subject 'Congratulations! Your Request has been accepted.' The email body reads: 'Hi There, Thanks for Registering as a Shelter Home in Make it Count. Hope you'll be satisfied with our Services. Warm Regards'. The email was sent at 4:08 PM (1 minute ago).

Approval Mail process builder

Process builder used to trigger the confirmation mail on status change is shown below,



Updated pending list for shelter homes

The admin can view all the pending list for shelter home before and after changing the “Shelter Status” value. This can be viewed in the report as shown below,

Shelter Home	Shelter Home Name	Email id	Address	Country
Shelter Status: Pending (3 records)				
New_Shelter_Home	-		New_Shelter_Home D03	Ireland
SH3	-		Delhi	India
SH2	-		test	Benin
Shelter Status: Approved (6 records)				
SH1	-		Test	Ireland
Dublin_Homeless_Shelters	admin@dubushomes.ie.com		Dublin_Homeless_Shelters, D 02	Ireland
Ireland_Homeless_Shelters	admin@rehomelessshelters.com.ie		Ireland_Homeless_Shelters, D 06	Ireland
Home_for_All	admin@homeforall.com.ie		Home_for_All, Dublin 07	Ireland
SH4	-		Pune	India
Challenge_Homelessness_Dublin	daniel@challengehgs.ie.com		Challenge_Homelessness, Dublin 02	Ireland
Grand Totals (9 records)				

Test Data:

We have created and entered data for Donor, Shelter Home, Request and Pledge creation to validate different scenarios.

- **Donor:** A donor can be an individual or an organization. The mandatory fields for donor creation are the name, e-mail id, password, country, and zip code. Entering the phone number is not mandatory. For each field, appropriate Salesforce field types are chosen and validated to make sure correct data are accepted before integrating it to the frontend. Below is an example where the password violates the validation rule, an error is thrown as the password length doesn't match the requirement. The screenshots related to front-end verification is added above in Donor account creation section. In that way, we have created multiple donors of both types.

The screenshot shows the 'Donor Edit' page with the following details:

- Header:** Donor Edit, New Donor, Save, Save & New, Cancel, Help for this Page.
- Error Message:** Error: Invalid Data. Review all error messages below to correct your data. The Password should contain at least one digit and one character; minimum length 6 characters.
- Information Section:** Donor Name: Jin Yang, Donor Type: Individual, Donor Email Id: jinyang@email.com, Password: 123#5 (highlighted in red), Donor Phone number: 9876543, Country: Antarctica, Currency: EUR - Euro.
- Buttons:** Save, Save & New, Cancel.

- **Shelter Home:** The Shelter home will have its name, e-mail id, password, address, phone number, country, and zip code. Same as above we have used random data to store and validate successful creation of a shelter home. In the example below a simple string of no email, the format has been entered and an error is thrown when we try to register it with that data.

The screenshot shows the 'Shelter Home Edit' page with the following details:

- Header:** Shelter Home Edit, New Shelter Home, Save, Save & New, Cancel, Help for this Page.
- Error Message:** Error: Invalid Data. Review all error messages below to correct your data.
- Information Section:** Shelter Home Name: O Connor's Shelter Home, Name: Connor, Email Id: connor45@gmail.com (highlighted in red), Password: 1234456, Phone Number: 8976543, Address: Dublin 17, Country: Ireland, Zip Code: 17, Shelter Status: Pending.
- Buttons:** Save, Save & New, Cancel.

- Pledge Creation: A pledge can be created by a donor only. To create a pledge the donor should choose a category from four options in the pick list, and for each category, there are different sub-categories available of which one needs to be picked. Optionally a user can choose a request Id to associate the pledge with a shelter home's request which is of type Campaign object in Salesforce. We have checked by selecting each category to make sure all the subcategory fields are being displayed accurately.

Pledge Edit
New Pledge

Pledge Edit Save Save & New Cancel

Information |= Required Information

Donor	Connor	Owner	Kunal Jangid
Currency	EUR - Euro		
Campaign	Winter Camp		
Category Level 1	Food		
Category Level 2	Available Frozen Food Cooked Food Fresh Food	Chosen 	
Status	Open		
Matched Request	<input type="text"/>		

Save Save & New Cancel

- Request Creation: Only shelter homes can create Requests. Same as a pledge the shelter home needs to choose category and subcategory which are mandatory for the creation of a request. Here the shelter home can pick an existing pledge id to associate their request. We have ensured that while creating all the categories and subcategories are in order and the Pledge list shows all the existing pledges to pick.

Search... Search

Signs Reports Donors Pledges Shelter Homes **Requests** Donor I

Request Edit Save Save & New Cancel

Information

Request Name	Request Decision	
Commodity Level 1	Clothes and Apparel	
Commodity Level 2	Available Blanket Rain gear	Chosen
Shelter Home	Dublin Homeless Shelters	
Campaign	Winter Camp	
Status	Open	
Match Pledge	<input type="text"/>	

Save Save & New Cancel

Search ~ Salesforce - Developer Edition - Google Chrome
https://eu16.salesforce.com/_ui/common/data/LookupPage?lkmf=editPage&lknm=CF00N1t00000L...

Lookup

Search... Go!

You can use '*' as a wildcard next to other characters to improve your search results.

Recently Viewed Pledges

Pledge Name
Pledge005
Pledge006
Pledge004
Pledge002
Pledge003
Pledge001

Copyright © 2000-2019 salesforce.com, inc. All rights reserved.

- Admin approval: Admin can approve a shelter home after verification. An email should be sent to the registered mail id upon a change in status. We have tested that by both accepting and rejecting approval requests to make sure the functionality is working perfectly.

The screenshot shows the Salesforce interface for editing a 'Shelter Home'. The top navigation bar includes links for Mail, Process Builders, Shelter Home, Salesforce, Truly shocking, Static Resources, Logo (200x200), (2) Visualforce, how to send, shelter hom, and other tabs. The main menu has options like Home, Dashboards, Campaigns, Reports, Donors, Pledges, Shelter Homes, Requests, Donor Registration, Donor Login, Pledge Creation, Shelter Home Registration, Shelter Home Login, Request Creation, and a Donation Management dropdown.

The current page is 'Shelter Home Edit' for the 'Homeless Welfare Association'. The 'Information' section contains the following data:

Field	Value
Shelter Home Name	Homeless Welfare Associa
Name	Harsh Jangra
Email Id	harshofficial@gmail.com
Password	*****
Phone Number	09969497570
Address	Dublin-02
Country	Ireland
Zip Code	2
Shelter Status	Pending --None-- Pending Approved

Buttons at the bottom include Save, Save & New, and Cancel. The page footer contains copyright information: Copyright © 2000-2019 salesforce.com, inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | 508 Compliance | Go to Salesforce mobile app.

Reporting and Dashboards

Operational Reports

Salesforce allows us to create several reports with the data in the system. There are several reports that will be used by the administrators regularly. For example, the report consisting a list of Shelter Homes with Pending Verification, as discussed above:

The screenshot shows a Salesforce report interface. At the top, there are two browser tabs: one for a pending verification report and another for a general shelter home report. The main window displays a report titled 'Shelter Status' with a time frame set to 'Custom' (Shelter Home: Created Date). The report is grouped by 'Shelter Status' and sorted by 'Shelter Status'. It lists two categories: 'Pending (3 records)' and 'Approved (6 records)'. The 'Pending' section includes entries for 'New Shelter Home', 'SH3', and 'SH2'. The 'Approved' section includes entries for 'SH1', 'Dublin Homeless Shelters', 'Ireland homeless shelters', 'Home for All', 'SH4', and 'Challenge Homelessness Dublin'. A summary at the bottom shows 'Grand Totals (9 records)'. The interface includes standard Salesforce navigation and reporting tools like 'Run Report', 'Save', and 'Printable View'.

Similarly, reports consisting list of 'Open' Requests and Pledges, along with their Category, will allow the administrators to easily match the requests and pledges.

The Report is scheduled to run daily using in-built Salesforce functionality. The updated reports including the latest data is used by admins, and also shared online with the wider community.

Open Request Categories

Schedule Report

Running User: Aditya Arora

Email Report: To me

Schedule Report

Frequency:

- Daily
- Weekly
- Monthly
- Every weekday
- Every day

Start: 20/04/2019

End: 20/05/2019

Preferred Start Time: Find available options...

Exact start time will depend on job queue activity.

Save Changes

- Save report modifications with this schedule
- Discard report modifications

Save Report Schedule | Cancel

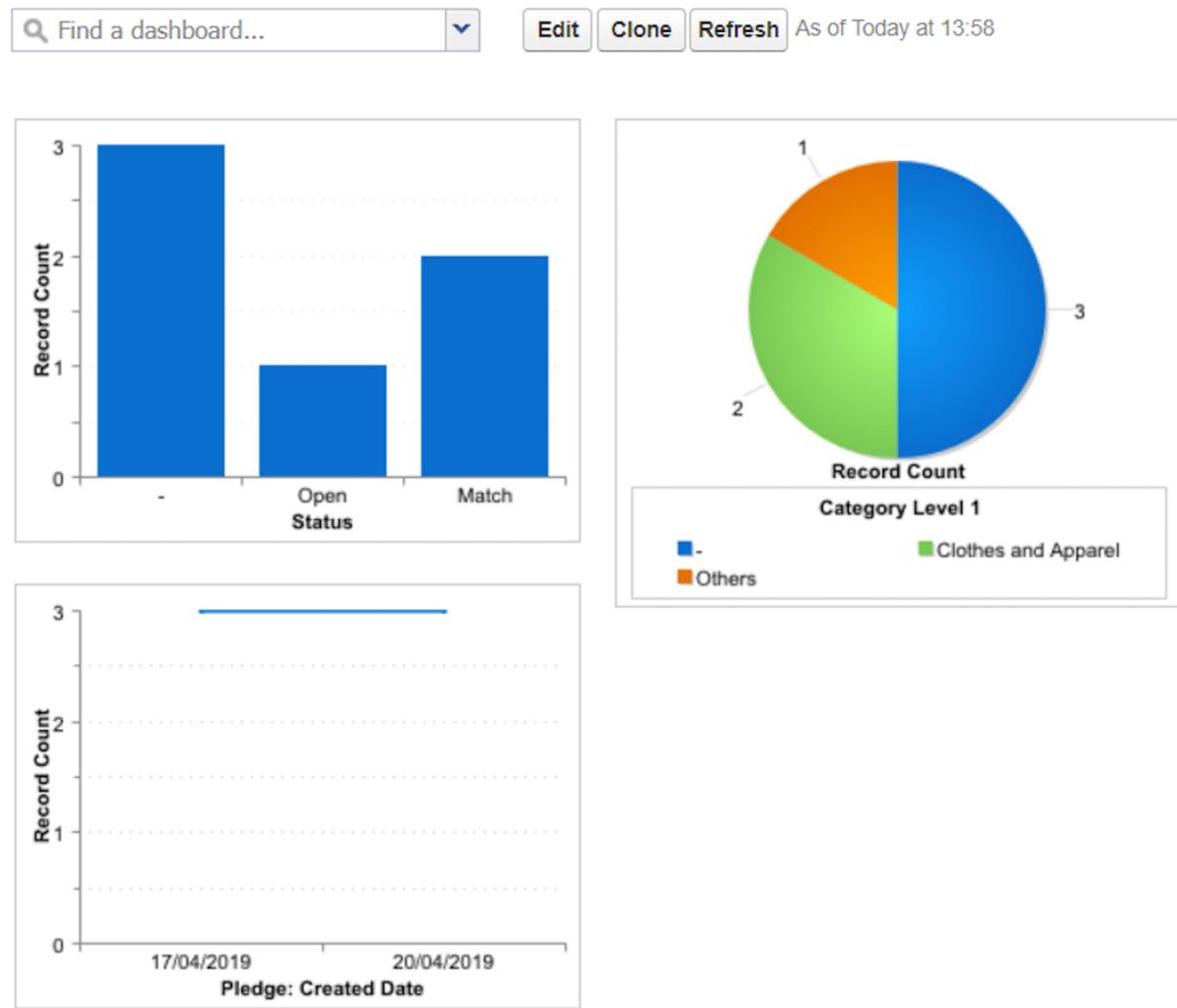
Management Dashboards

Apart from these transactional reports, there are various Management Reports, arranged into Dashboards, to provide a summary of the application.

Dashboard 1: Overview

[« Go to Dashboard List](#)

Pledges - Management Summary



The Overview Dashboard contains various reports providing important metrics regarding the application usage.

The reports help us analyse:

1. Top donors based on number of pledges created
2. Top shelter homes based on number of requests created

3. Number of Shelter Homes with pending verification vs number of verified Shelter Homes
4. Distribution of donors by Individual and Organization Donors

Such details are important for Customer Management and Donor Management. For example, top Donors can be identified and personally acknowledged, which will help retain them on the platform.

Dashboard 2: Pledge Overview

[« Go to Dashboard List](#)

Overview



This dashboard provides a brief summary of pledges logged in the system.

The above reports provide important data including:

1. Number of Matched vs Open Pledges
2. Trend of Pledges created in the current month
3. Categories Distribution for Open Pledges

These reports help identify trend of the pledges created. A systematic decrease in pledges can be detected and followed up by the management.

Dashboard 3: Request Overview

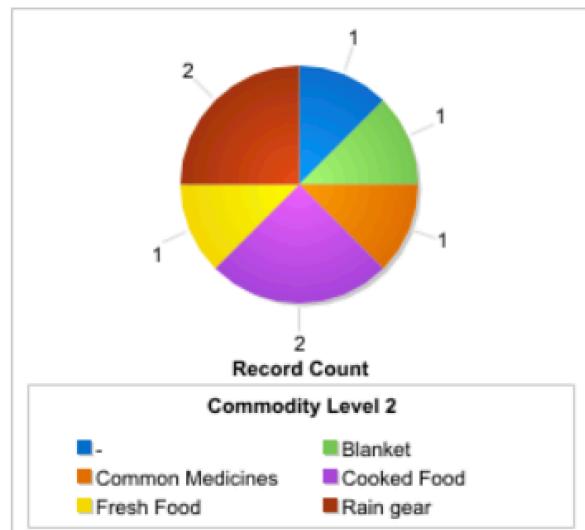
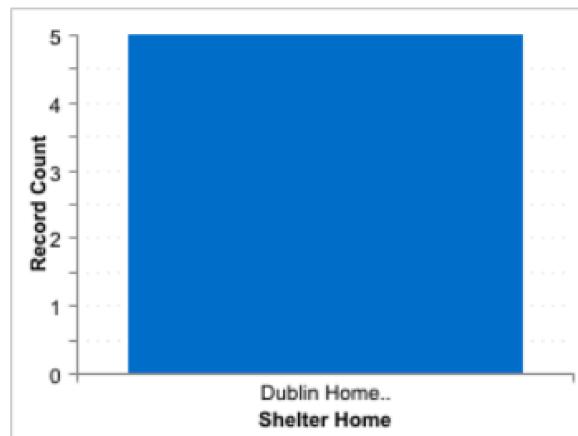
[« Go to Dashboard List](#)



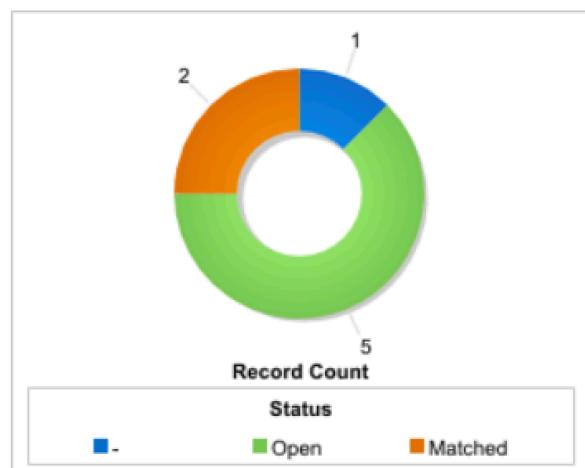
Requests Management Summary

Find a dashboard... Edit Clone Refresh As of Today at 13:48

Requests Management Summary



Commodity Level 2	Record Count
Blanket	1
Common Medicines	1
Cooked Food	1
Fresh Food	1
Rain gear	1



The dashboard provides a detailed summary of the Requests created in the system.

The reports provide information regarding:

1. Category distribution of Open Requests
2. Number of Matched vs Open Requests
3. Number of Open Requests by Shelter Home
4. Number of Open Requests for each Category

These reports help us analyse which categories are more widely requested by shelter homes. This can help the platform organize targeted campaigns to satisfy the requirements.

References

DATA THAT CHANGES THE WORLD, eBook by bloomerang.co. Available at: <https://bloomerang.co/images/uploads/The-Ultimate-Guide-to-Nonprofit-Donor-Data-Management.pdf>

NGO Donation collection Process Flow |Published by BluSynergy on Mar 11, 2015. Available at: https://www.slideshare.net/BluSynergy_India/ngo-process-flow

Build vs. Buy: How to Know When You Should Build Custom Software Over Canned Solutions, Contributor Chuck Cohn | Published by Forbs on Sep 15, 2014. Available at: <https://www.forbes.com/sites/chuckcohn/2014/09/15/build-vs-buy-how-to-know-when-you-should-build-custom-software-over-canned-solutions/> - 46a2b32fc371

20 Best Donor Management Software Solutions of 2019, By: FinancesOnline Available at: <https://financesonline.com/top-20-donor-management-software-solutions/Best>

Donor Management Software, By: G2, Available at: <https://www.g2.com/categories/donor-management>

Appendix

// Visualforce pages

//Welcome Page

```
// Welcome Page
<apex:page showHeader="false" sidebar="false" standardStylesheets="true"
controller="WelcomeController">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-----Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }
</style>
<!-----Style Ends----->
<!--script start-->
<script>
```

```

        function clearValue() {
            document.getElementById('id$=j_id0:j_id3:input-id-001').value = '';
        }
    <!--script end----->
<div align ="center">
<apex:image url="https://c.eu16.visual.force.com/resource/1555766834000/Logo"
styleClass="imageStyle" style="margin-left:50%,margin-right:-10%"/> </div>
<div align ="center">
<apex:form style="margin-left:30%;margin-right:-30%;">
<div class="container">
    <div class="slds-card_header">
        <p style="font-size:20px;text-align:center;">Select your Member Type</p>
    </div>
    <div class="slds-form-element">
        <apex:commandButton styleClass="slds-button slds-button--brand btn-lg"
value="Donor" action="!donorPage"/>
        &nbsp;
        <apex:commandButton styleClass="slds-button slds-button--destructive btn-lg"
value="Shelter Home" action="!shelterPage"/>
    </div>
</div>
</apex:form>
</div>
</apex:page>

```

//Shelter Home Registration

```

<apex:page standardStylesheets="false" showHeader="false"
standardController="Shelter_Home_c" lightningStylesheets="true"
extensions="Registrationfromcontroller">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!--Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

```

```

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }

```

</style>

<!--Style Ends-->

<apex:form style="margin-left:30%;margin-right:-30%;">

<div class="container">

 <div class="slds-card_header">

 <p style="font-size:20px;text-align:center;">Shelter Home Registration Form</p>

 </div>

 <div class="slds-form-element">

 <label class="slds-form-element_label" for="input-id-01">Full Name</label>

 <div class="slds-form-element_control">

 <apex:inputField id="input-id-01" value="{!!acco.Name__c}" styleClass="slds-input" html-placeholder="Full Name"/>

 </div>

 </div>

 <div class="slds-form-element">

 <label class="slds-form-element_label" for="input-id-01">Phone</label>

 <div class="slds-form-element_control">

 <apex:inputField id="input-id-02" value="{!!acco.Phone_Number__c}" styleClass="slds-input" html-placeholder="Phone No"/>

 </div>

 </div>

 <div class="slds-form-element">

 <label class="slds-form-element_label" for="input-id-01">Email</label>

 <div class="slds-form-element_control">

 <apex:inputField id="input-id-03" value="{!!acco.Email_id__c}" styleClass="slds-input" html-placeholder="Email"/>

 </div>

 </div>

 <div class="slds-form-element">

 <label class="slds-form-element_label" for="input-id-01">Password</label>

 <div class="slds-form-element_control">

 <apex:inputSecret lang="*" id="input-id-04" value="{!!acco.Password__c}" styleClass="slds-input" html-placeholder="Password"/>

 </div>

 </div>

 <div class="slds-form-element">

 <label class="slds-form-element_label" for="input-id-01">Address</label>

 <div class="slds-form-element_control">

 <apex:inputField id="input-id-05" value="{!!acco.Address__c}" styleClass="slds-input" html-placeholder="Address"/>

 </div>

 </div>

 <div>

 </div>

 <div>

 <apex:commandButton styleClass="slds-button slds-button_brand" value="Create Account" action="{!!save}"/>

 <apex:commandButton styleClass="slds-button slds-button_destructive" value="Cancel" action="{!!cancel}"/>

```

        </div>
<div>
    &nbsp;
        <p>Are you a Registered Shelter Home? <a href="https://c.eu16.visual.force.com/apex/ShelterHomeReg">Register Here</a></p>
    </div>
</div>
</apex:form>
</apex:page>

```

//Shelter Log in

```

<apex:page showHeader="false" sidebar="false" standardStylesheets="true"
controller="ShelterLoginClass">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-----Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }
</style>

```

```

<!-----Style Ends----->
<!--script start---->
<script>
    function clearValue() {
        document.getElementById('id$j_id0:j_id3:input-id-001').value = '';
    }
</script>
<!--script end----->

<apex:form style="margin-left:30%;margin-right:-30%;">
<div class="container">
    <div class="slds-card_header">
        <p style="font-size:20px;text-align:center;">Shelter Login Page</p>
    </div>
    <div class="slds-form-element">
        <label class="slds-form-element_label" for="input-id-01">User Name</label>
        <div class="slds-form-element_control">
            <apex:inputText id="input-id-001" value="{!userName1}" styleClass="slds-input" html-placeholder="UserName"/>
        </div>
    </div>
    <div class="slds-form-element">
        <label class="slds-form-element_label" for="input-id-01">Password</label>
        <div class="slds-form-element_control">
            <apex:inputSecret lang="*" id="input-id-02" value="{!password}" styleClass="slds-input" html-placeholder="Password"/>
        </div>
    </div>
    <div>
        &nbsp;
    </div>
    <div>
        <apex:pageMessages ></apex:pageMessages>
    </div>
    <div>
        <apex:commandButton styleClass="slds-button slds-button_brand" value="Login" action=" {!verifyShelterHome}"/>
        &nbsp;
        <apex:commandButton styleClass="slds-button slds-button_destructive" value="Reset" action=" {!restbutton}"/>
    </div>
    <div>
        &nbsp;
        <p>Are you a Register Shelter home? <a href="https://c.eu16.visual.force.com/apex/ShelterHomeReg">Register Here</a></p>
    </div>
</div>
</apex:form>
</apex:page>

```

//Request Creation

```

<apex:page standardStylesheets="false" showHeader="false" standardController="Request__c" lightningStylesheets="true">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!--Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

```

```

.container {
    border: 2px solid #dedede;
    background-color: #f1f1f1;
    border-radius: 5px;
    padding: 10px;
    margin: 10px 0;
}

.darker {
    border-color: #ccc;
    background-color: #ddd;
}

.container::after {
    content: "";
    clear: both;
    display: table;
}

.container img {
    float: left;
    max-width: 60px;
    width: 100%;
    margin-right: 20px;
    border-radius: 50%;
}

.container img.right {
    float: right;
    margin-left: 20px;
    margin-right:0;
}

.time-right {
    float: right;
    color: #aaa;
}

.time-left {
    float: left;
    color: #999;
}

</style>
<!--Style Ends-->
<apex:form style="margin-left:30%;margin-right:-30%;">
<div class="container">
    <div class="slds-card__header">
        <p style="font-size:20px;text-align:center;">Create Your Request</p>
    </div>
    <div class="slds-form-element">
        <label class="slds-form-element__label" for="input-id-01">Pledge Name</label>
        <div class="slds-form-element__control">
            <apex:inputField id="input-id-03" value="{!Request__c.Match_Pledge__c}" styleClass="slds-input" html-placeholder="Request Name"/>
        </div>
    </div>
    <div class="slds-form-element">
        <label class="slds-form-element__label" for="input-id-01">Category</label>
        <div class="slds-form-element__control">
            <apex:inputField id="input-id-01" value="{!Request__c.Commodity_Level_1__c}" styleClass="slds-input" html-placeholder="Category"/>
        </div>
    </div>
    <div class="slds-form-element">
        <label class="slds-form-element__label" for="input-id-01">Items</label>
        <div class="slds-form-element__control">

```

```

        <apex:inputField id="input-id-02"
value="{!!Request__c.Commodity_Level_2__c}" styleClass="slds-input" html-
placeholder="Items"/>
    </div>
</div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">Campaign Name</label>
    <div class="slds-form-element__control">
        <apex:inputSecret lang="" id="input-id-04" value="{!!Request__c.Campaign__c}" styleClass="slds-input" html-placeholder="Campaign Name"/>
    </div>
</div>
<div>
    &nbsp;
</div>
<div>
    <apex:commandButton styleClass="slds-button slds-button_brand" value="Create Pledge" action=" {!save} "/>
    &nbsp;
    <apex:commandButton styleClass="slds-button slds-button_destructive" value="Cancel" action=" {!cancel} "/>
</div>

</div>
</apex:form>
</apex:page>

```

//Pledge Creation

```

<apex:page standardStylesheets="false" showHeader="false"
standardController="Pledge__c" lightningStylesheets="true">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-----Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

```

```

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }

```

</style>

<!--Style Ends-->

<apex:form style="margin-left:30%;margin-right:-30%;">

<div class="container">

 <div class="slds-card__header">
 <p style="font-size:20px;text-align:center;">Create Your Pledge</p>
 </div>

 <div class="slds-form-element">
 <label class="slds-form-element__label" for="input-id-01">Request Name</label>
 <div class="slds-form-element__control">
 <apex:inputField id="input-id-03" value="{!Pledge__c.Matched_Request__c}" styleClass="slds-input" html-placeholder="Request Name"/>
 </div>
 </div>

 <div class="slds-form-element">
 <label class="slds-form-element__label" for="input-id-01">Category</label>
 <div class="slds-form-element__control">
 <apex:inputField id="input-id-01" value="{!Pledge__c.Pledge_Category_Level_1__c}" styleClass="slds-input" html-placeholder="Category"/>
 </div>
 </div>

 <div class="slds-form-element">
 <label class="slds-form-element__label" for="input-id-01">Items</label>
 <div class="slds-form-element__control">
 <apex:inputField id="input-id-02" value="{!Pledge__c.Category_Level_2__c}" styleClass="slds-input" html-placeholder="Items"/>
 </div>
 </div>

 <div class="slds-form-element">
 <label class="slds-form-element__label" for="input-id-01">Campaign Name</label>
 <div class="slds-form-element__control">
 <apex:inputSecret lang="*" id="input-id-04" value="{!Pledge__c.Campaign__c}" styleClass="slds-input" html-placeholder="Campaign Name"/>
 </div>
 </div>

 <div>

 </div>

 <div>
 <apex:commandButton styleClass="slds-button slds-button_brand" value="Create Pledge" action=" {!save} "/>

 <apex:commandButton styleClass="slds-button slds-button_destructive" value="Cancel" action=" {!cancel} "/>
 </div>

</div>

</apex:form>

</apex:page>

//Donor Registration

```
<apex:page showHeader="false" sidebar="false" standardStylesheets="true"
controller="Loginverificationclass">
<apex:slds />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-----Style Start----->
<style>
    body {
        margin: 0 auto;
        max-width: 800px;
        padding: 0 20px;
    }

    .container {
        border: 2px solid #dedede;
        background-color: #f1f1f1;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
    }

    .darker {
        border-color: #ccc;
        background-color: #ddd;
    }

    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    .container img {
        float: left;
        max-width: 60px;
        width: 100%;
        margin-right: 20px;
        border-radius: 50%;
    }

    .container img.right {
        float: right;
        margin-left: 20px;
        margin-right:0;
    }

    .time-right {
        float: right;
        color: #aaa;
    }

    .time-left {
        float: left;
        color: #999;
    }
</style>
<!-----Style Ends----->
<!--script start-->
<script>
    function clearValue() {
        document.getElementById('id$j_id0:j_id3:input-id-001').value = '';
</script>
<!--script end-->

<apex:form style="margin-left:30%;margin-right:-30%;">
<div class="container">
    <div class="slds-card__header">
```

```

        <p style="font-size:20px;text-align:center;">Login Page</p>
    </div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">User Name</label>
    <div class="slds-form-element__control">
        <apex:inputText id="input-id-001" value="{!userName1}" styleClass="slds-input" html-placeholder="UserName"/>
    </div>
</div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-01">Password</label>
    <div class="slds-form-element__control">
        <apex:inputSecret lang="" id="input-id-02" value="{!password}" styleClass="slds-input" html-placeholder="Password"/>
    </div>
</div>
<div class="slds-form-element">
    <label class="slds-form-element__label" for="input-id-03">Donor Type</label>
    <div class="slds-form-element__control">
        <apex:selectList id="Donortype" value="{!donortype}" size="1">
            <apex:selectOption itemValue="Individual" itemLabel="Individual"/>
            <apex:selectOption itemValue="Organization" itemLabel="Organization"/>
        </apex:selectList>
    </div>
</div>
<div>
    &nbsp;
</div>
<div>
    <apex:pageMessages ></apex:pageMessages>
</div>
<div>
    <apex:commandButton styleClass="slds-button slds-button_brand" value="Login" action="{!verifydonor}"/>
    &nbsp;
    <apex:commandButton styleClass="slds-button slds-button_destructive" value="Reset" action="{!restbutton}"/>
</div>
<div>
    &nbsp;
    <p>Are you a Register Donor? <a href="https://c.eu16.visual.force.com/apex/RegistrationPage?sfdc.tabName=01r1t000000AV15">Register Here</a></p>
</div>
</div>
</apex:form>
</apex:page>

```

```

/* Apex Classes -
Standard and Extended controllers for all pages */
Apex Controller
// Apex Controller for Welcome Page
public class WelcomeController {

    public pageReference donorPage{
        Pagereference pg = new Pagereference('/apex/UserLoginPage_VF');
        pg.setredirect(true);
        return pg;
    }
}

```

```

        }
    public pageReference shelterPage
    {
        Pagereference pg = new Pagereference('/apex/ShelterLogin');
        pg.setredirect(true);
        return pg;
    }
// Apex Controller for Donor Registration
public class Registrationfromcontroller {

    public Donor__c acco{get; set; }

    public Registrationfromcontroller(ApexPages.StandardController controller)
    {
        acco=new Donor__c();
    }

    public pagereference Save()
    {

        try{
            insert acco;
            string [] emailaddress= new String[]{acco.Donor_Email_Id__c};
            sendemail(emailaddress);
            Pagereference pg = new Pagereference('/apex/AccountiscreatedPage');
            pg.setredirect(true);
            return pg;
        }
        catch(exception e)
        {
            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.WARNING,'Account No Created, Please check data'));
        }
        return null;
    }

    Public void sendemail(string[] email)
    {
        Messaging.SingleEmailMessage semail = new Messaging.SingleEmailMessage();
        semail.setToAddresses(email);
        semail.setSubject('Congratulations, you have successfully registered!');
        semail.setPlainTextBody('Hi There, Welcome to the Make it Count Family. Thanks for your prestigious contribution towards our collaborated goal to eradicate homelessness and poverty.');
        Messaging.sendEmail(new Messaging.SingleEmailMessage[] {semail});
    }

}

```

```

// login Verification Class
public class Loginverificationclass
{
    public string userName1 {get;set;}
    Public string password {get;set;}
    public string donortype{get;set;}
    public boolean flag=false;
    public Donor__c acco{get; set; }

    public PageReference verifydonor()
    {
        list<Donor__c> acco=[select id,Donor_Email_Id__c,Password__c from Donor__c];
        for(Donor__c donor:acco)
        {
            if(donor.Donor_Email_Id__c==userName1 && donor.Password__c==password)
            {

```

```

        flag=true;
    }
}
if(flag==true)
{
    Pagereference pg = new Pagereference('/apex/PledgeVF');
    pg.setredirect(true);
    return pg;
}
else
{
    ApexPages.addmessage(new ApexPages.message(ApexPages.severity.WARNING,'
You are not Registered Donor'));
    userName1=null;
}
return null;
}

public void restbutton()
{
    userName1=null;
    password=null;
}
}

// Registration Form Controller
public class Registrationfromcontroller {

    public Donor__c acco{get; set; }

    public Registrationfromcontroller(ApexPages.StandardController controller)
    {
        acco=new Donor__c();
    }

    public pagereference Save()
    {

        try{
            insert acco;
            string [] emailaddress= new String[]{acco.Donor_Email_Id__c};
            sendemail(emailaddress);
            Pagereference pg = new Pagereference('/apex/AccountiscreatedPage');
            pg.setredirect(true);
            return pg;
        }
        catch(exception e)
        {
            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.WARNING,'
Account No Created, Please check data'));
        }
        return null;
    }

    Public void sendemail(string[] email)
    {
        Messaging.SingleEmailMessage semail = new Messaging.SingleEmailMessage();
        semail.setToAddresses(email);
        semail.setSubject('Congratulations, you have successfully registered!');
        semail.setPlainTextBody('Hi There, Welcome to the Make it Count Family. Tha
nks for your prestigious contribution towards our collaborated goal to eradicate ho
melessness and poverty.');
        Messaging.sendEmail(new Messaging.SingleEmailMessage[] {semail});
    }
}
}

```

```

//Shelter Login Class
public class ShelterLoginClass {
    public string userName1 {get;set;}
    Public string password {get;set;}
    public boolean flag=false;
    public Shelter_Home__c acco{get;set;}

    public PageReference verifyShelterHome()
    {
        list<Shelter_Home__c> acco=[select id,Email_Id__c,Password__c from
Shelter_Home__c];
        for(Shelter_Home__c shelterhome:acco)
        {
            if(shelterhome.Email_Id__c==userName1 &&
shelterhome.Password__c==password)
            {
                flag=true;
            }
        }
        if(flag==true)
        {
            Pagereference pg = new Pagereference('/apex/RequestVF');
            pg.setredirect(true);
            return pg;
        }
        else
        {
            ApexPages.addmessage(new
ApexPages.message(ApexPages.severity.WARNING,'You are not Registered Donor'));
            userName1=null;
        }
        return null;
    }

    public void restbutton()
    {
        userName1=null;
        password=null;
    }
}

```