**SOLUTION DOCUMENT OF NEW LOS**

Table of Contents

[**1.** **Login** 4](#_Toc189231665)

[**1.1** **Login Functionality** 4](#_Toc189231666)

[**1.2** **Forgot Password Functionality** 4](#_Toc189231667)

[**1.3** **Change Password Functionality** 4](#_Toc189231668)

[**2.** **Index** 6](#_Toc189231669)

[**2.1** **Second Row Actions** 6](#_Toc189231670)

[**2.2** **Bank Process Button Click** 7](#_Toc189231671)

[**2.3** **Borrower & Co-Borrower Documents** 8](#_Toc189231672)

[**2.4** **Tabs at Bottom** 8](#_Toc189231673)

[**2.5** **Side Buttons** 8](#_Toc189231674)

[**3.** **Index design** 9](#_Toc189231675)

[**4.** **Post Sanction** 12](#_Toc189231676)

[**5.** **Post Sanction Design** 12](#_Toc189231677)

[**6.** **Ready To Audit** 13](#_Toc189231678)

[**7.** **Neft Done** 14](#_Toc189231679)

[**8.** **Common Design** 14](#_Toc189231680)

[**9.** **Master’s** 16](#_Toc189231681)

[**9.3** **End Points** 16](#_Toc189231682)

[**9.4** **Add** **Employee** 17](#_Toc189231695)

[**9.5** **Add Creator** 17](#_Toc189231696)

[**9.6** **Add Branch** 17](#_Toc189231697)

[**9.7** **Add Group** 17](#_Toc189231698)

[**9.8** **Vehicle Master** 18](#_Toc189231699)

[**9.9** **Add Banner** 18](#_Toc189231700)

* **Dashboard**
* **Master**  
  -Menu [Add menu / permissions]  
   -Roles [dept /Designation/ Role]  
   -Creator  
   -Branch  
   -Vehicle [fuel/ brand/Model/ Vehicle]

- End Points

* **FI Processing**
* **Credit**  
   -Modification [before after Sanction amount / Disbursement date / Loan closure Amount]  
   -Generate [Cam generation / Disbursement Sheet / Download One Pager]  
   -Delete [ Crif / 2nd e-sign / RC / Docs]  
   -Docs Upload [ upload docs/ Delete docs req]  
   -PDD Documents  
   -Index
* **Audit**  
  **-**Ready to Audit
* **Account**  
   -Ready for NEFT
* **Branch**  
   -Add Guarantor  
   -Asset Request  
   -Guarantor Crif  
   -Post Sanction
* **HRMS**
* **HRMS**  
   -Asset Approval

-Asset Dispatched

1. **Login**

# **Login Functionality**

* **Objective**: Allow users to log in using their User ID and Password.
* **Process**:
  + Upon successful validation of the provided credentials (User ID and Password), a JWT (JSON Web Token) is generated.
  + The data used to generate the JWT token includes:
    - **Id**, **Name**, **Email**, **Role**, **Employee Code**, and **Creator**.
  + The **JWT token** is returned to the client and will be used for authenticating future requests across the application.
  + The user is redirected to the dashboard upon successful login.
  + The system only allows access to users whose **IsActive** status is set to 1 in the **EmployeeDetails** table.

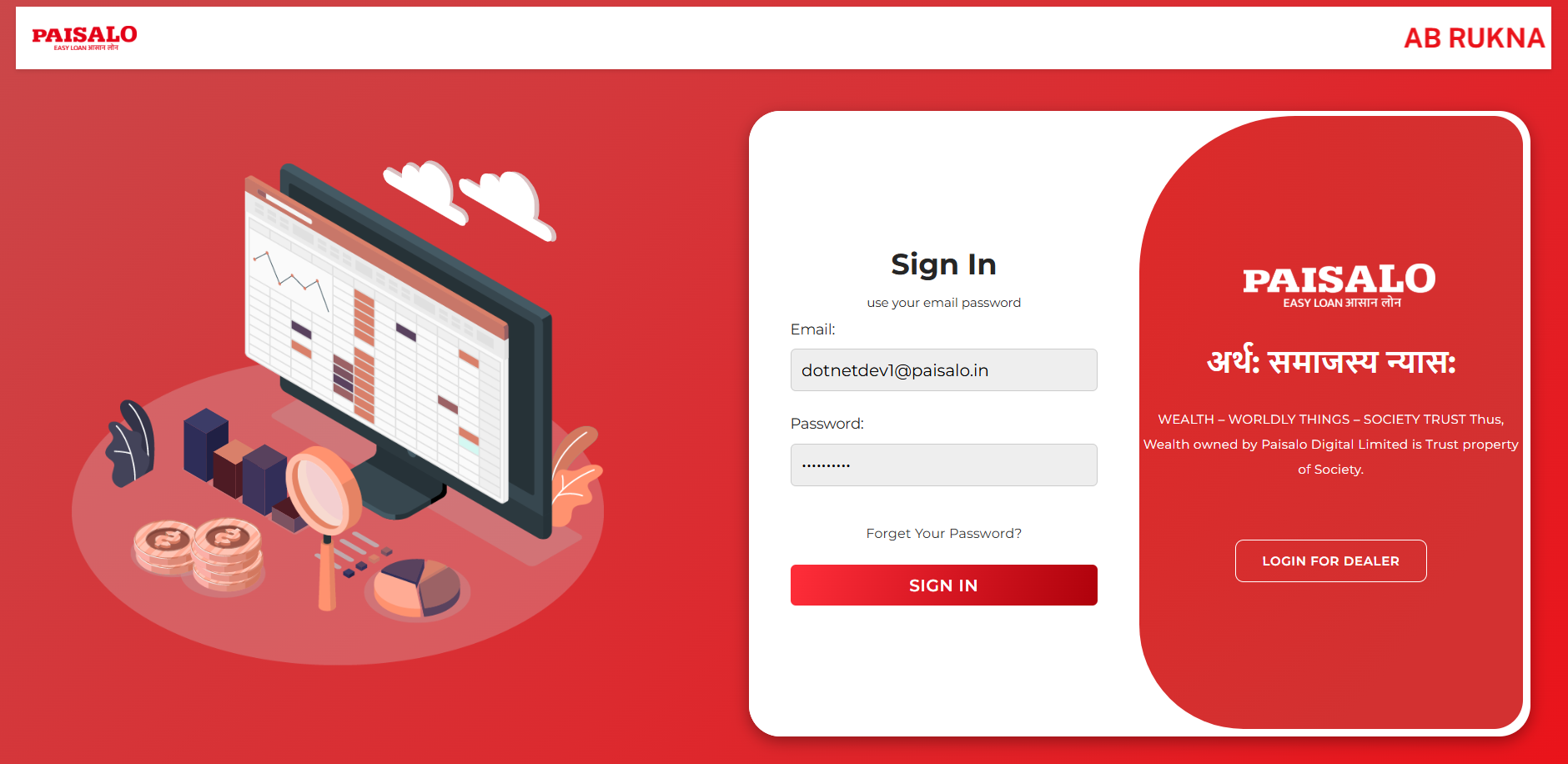
# **Forgot Password Functionality**

* **Objective**: Allow users to reset their password if they forget it.
* **Process**:
  + When the user clicks on the **Forgot Password** option, the system generates a **6-digit code**.
  + This code is saved in local memory (temporarily) and sent to the user’s registered email address.
  + The user is then prompted to enter the **6-digit code**.
  + The system validates the entered code against the stored code. If they match, the user is given the option to **enter a new password**.
  + After entering and confirming the new password, the password is updated in the database.

# **Change Password Functionality**

* **Objective**: Allow users to change their current password.
* **Process**:
  + When the user clicks on the **Change Password** option, three textboxes will appear:
    - **Old Password**, **New Password**, and **Re-enter New Password**.
  + The system first validates the **Old Password** by comparing it with the stored password in the database.
  + If the old password is valid, the user can proceed to enter a **New Password** and **Re-enter New Password** to confirm it.
  + If both new passwords match, the password is updated in the database.

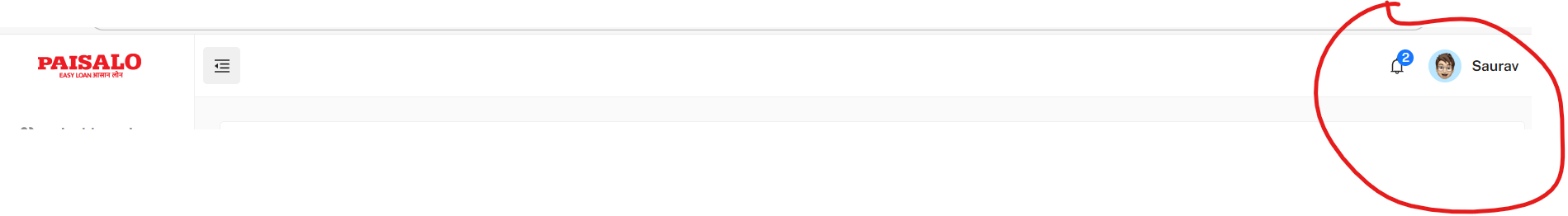
**Design**:



-Add Change password option

-Add Loader where required

**Top Bar:**

****

**-**At top right Conner login user details should bind dynamically. [User name, Role. Logout]

-On logout click redirect back on login UI

**Menu:**

-Maintain the hierarchy in design, it should visible like

* Main Menu1

Sub Menu

Related UI Page1

Related UI Page2

Related UI Page3.

* Main Menu2

Sub Menu

Related UI Page1

Related UI Page2

Related UI Page3

# **Index**

* Bind Creator Dropdown (All in Capital Letters, No Duplication, Search on Key Press.
* The user can search either **Creator and** **Ficode** or **Smcode**.
* Display basic information (already stored in the database) once the user performs a search. There is no case when no record will be found.

# **Second Row Actions**

* **Options**:
  + Account opening
  + Loan Agreement (2nd esign document download)
  + House Visit
  + Generate CRIF (If CRIF (valid for 30 days) already exists, display it. Otherwise, show a Generate CRIF button to create a new CRIF report.)
  + PDD Document (Only visible in VH case)
  + Bank Process

# **Bank Process Button Click**

When the **Bank Process Button** is clicked, the system should first perform a series of related validations. These validations must ensure that all necessary data is present in the corresponding **data tables**, which should include personal information, guarantor details, bank details, contact details, and other related data. Additionally, the system must check that personal details match with the CRIF generation details, and verify documents, IDs, date of birth (DOB), age, etc.

After these validations are complete, any missing or mismatched information should be noted. This missing or mismatched data should then be returned via an **API** response, which will contain all the necessary details. The response should be displayed in a **popup on the UI**, allowing the user to easily identify the missing parts

The next set of validations involves ensuring that all necessary details are verified and compliant before proceeding with the bank process. First, the system must verify that all required **FI documents** [ **CheckDocumentsAndReturnStatus]**. This will ensure that all documents are present and valid for processing. Secondly, the system must check that essential **bank details** such as the **Bank Name**, **IFSC**, **Account Number**, and **Bank Address** are not **null**. If any of these fields are missing, the process cannot continue. Finally, a **religion-based validation** must be implemented. If the borrower’s religion is **Muslim**, the **CRIF Score** must be greater than **715** to proceed. If the CRIF score exceeds 715, an additional check is required: the religion of the related group code FI must be reviewed. If all members of the related FI group are **Muslim**, the case must be **rejected**. These validations ensure the integrity of the bank process while adhering to specific business and compliance rules.

In the case that the **Ficode (Ficase)** is unable to proceed with the **bank process** due to missing or incorrect information, the entry should be saved in the **FiJourney** table, and the missing or erroneous details should be logged into the **Error\_log** field with the status set to "Bank Process Click." This ensures that all issues are tracked and can be reviewed for corrective actions.

If all the validations are successfully passed, the system will insert the validated details into the **FiForBankProcess** table, assigning the **Highest Priority Bank ID** and setting the **Status** to **0** by default. Once the details are inserted, a **backend job** will be triggered to send these cases to different banks for **sanctioning**. As the process progresses, logs will be maintained to track the status of the cases. Once the sanction is received from the banks, the system will update the **FiCurrentStatus** table, setting the **status code** to **6** to indicate that the sanction process has been completed. This ensures that all steps are tracked.

# **Borrower & Co-Borrower Documents**

Functionality involves retrieving and displaying all related documents for both the borrower and co-borrower. When the user clicks on **View Document**, the system will fetch all the documents linked to the borrower and co-borrower from the database. Upon clicking the **Download** button, the respective document will be available for download. The system should visually highlight any **missing documents** by marking them with a **red colour**. This helps to easily identify any missing documents, prompting the user to take action and upload them before proceeding.

# **Tabs at Bottom**

**General Details:**

**Personal Details**

**Financial Details**

**Co-borrower Details**

**Top-up Details**

**Other Links:**

**1.Geo Location**: Geo-tagging feature to display a map view showing all locations based on their latitude and longitude coordinates when FO visit borrower house.

**2.Pronote:** Download document

**3.Clone Status Button** functionality begins by validating the entered **Aadhaar number** to check if any previous loan details exist in the database. If a match is found, the system will retrieve the related loan details associated with that Aadhaar number. These details will then be **bound to the UI** and displayed in a **popup**, allowing the user to review all the previous loan information.

**4.Account Aggregator:** Download RDLC Report

# **Side Buttons**

**Add**: **Add** **New FI** trigger the creation of a new **FI entry** populated with all the existing details from the current record. This new entry will contain the same data as the previous one but will be assigned a new **FI\_ID**.

Upon creating the new **FI entry**, the following steps should be carried out:

* + 1. The **IsActive** field for the previous **FI** should be updated to **0**, indicating that it is no longer active.
    2. The **FI\_ID** of the previous record should be updated in all related tables to ensure consistency and maintain proper references across the system.

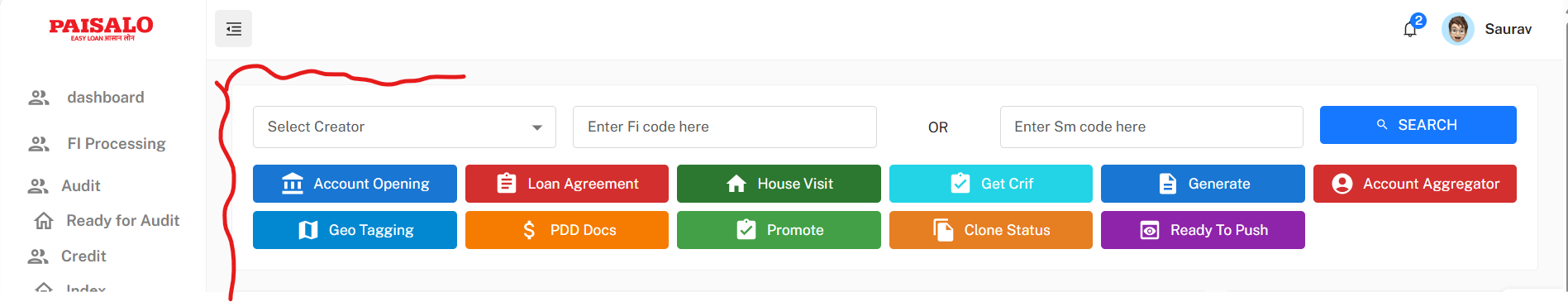
**Edit**:

Case 1> Before 1st e-sign [**FiUploadBeforeFirstSign]**  
Before the first e-sign, allow editing of the following fields: Name, Father's Name, Gender, Date of Birth, Region, Caste, Loan Amount, Loan Duration, Loan Reason, Pin code, Address 1, 2, 3 (Aadhar Address), Alternate Phone Number, and Mobile Number. Display the Borrower Score if it exists in the table; otherwise, generate it using Borrower Score rules engine. If case is created by a VH Creator, show Dealer, Vehicle, and Model Details.  
  
Case 2> Before sanction [**UpdateBeforeSanction]**  
Editable fields Loan amount, duration, reason, bank Acc, IFSC,   
co-borrower details  
  
Case 3> Before 2nd e-sign **[UpdateBeforeSecondESign]**Editable fields Account no, IFSC

**Update:** After Edit the details

**Delete**: Update IsActive-0 and is delete =1

# **Index design**

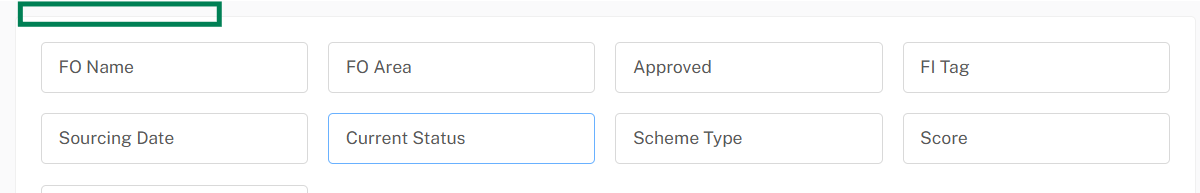


-decrease this white space, add breadcrumbs here.

-1st row, creator dropdown, textbox for fiCode, textbox for search box, Search button

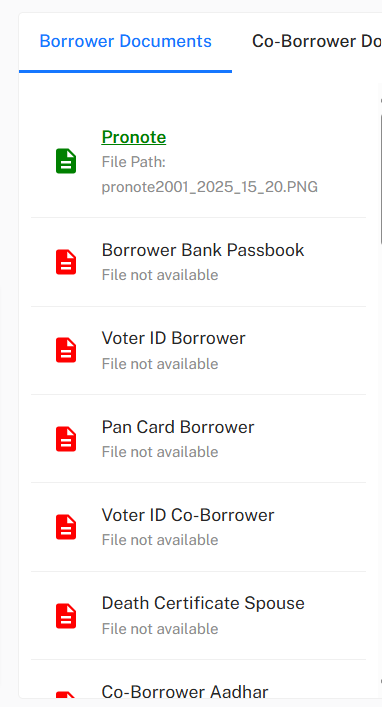
-2nd row, Account opening, Loan Agreement, House Visits, Crif, PDD Docs, Bank Process

-3rd section

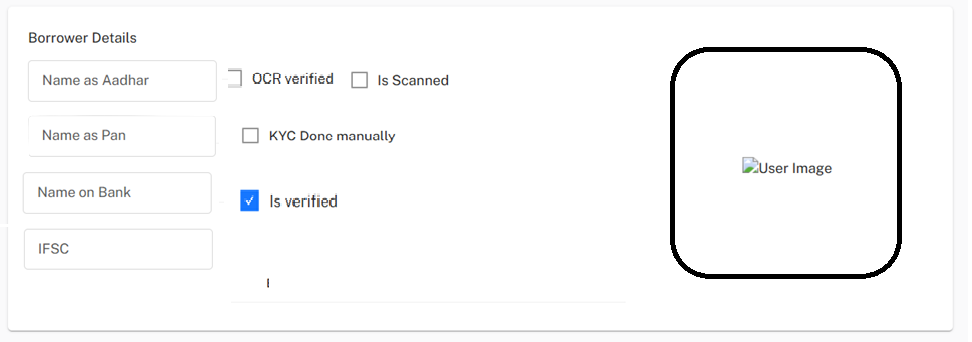
****

Display FoName, Area, Tag, Sourcing date, current status, schemcode as label. At top where green box is visible add Text Approved/ Not Approved (Approved in green, not approved in red)

- Section borrower and Co-Borrower, on page load static design should be there no any empty space should visible even after search when no record found.



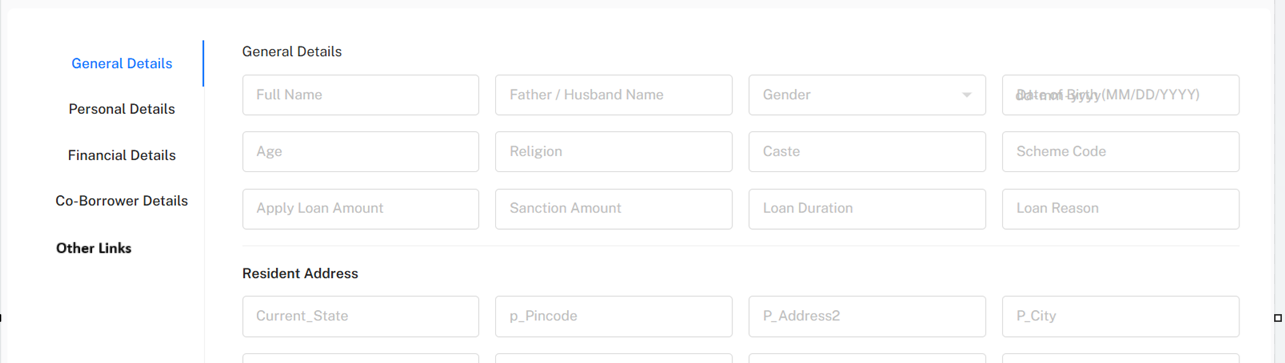
-4th section, Borrower Details



Is profile pic at right not coming and on page load then any static image should visible.

Use labels for view data, not in text box.

-5th section links for general details



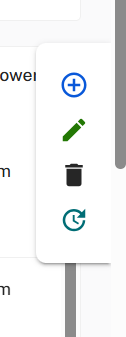
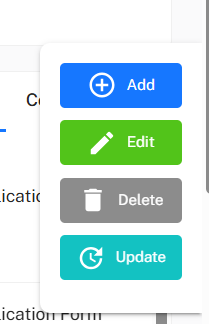
-Use loader after search

-On clone statue click, open popup with details [ Smcode, fiCode, creator, current status, Amount, closing date], there is no any previous details the show proper msg.

- on click Bank Process button, if any validation msg return from API’s then show those messages in Popup.

-loader on search click

-Side buttons, use more attractive Icons

If user click on Edit, focus shift on editable textboxes.

On update click, loader is required.

On Add Click, reload page and automatically searched with new Ficode, creator and display msg.

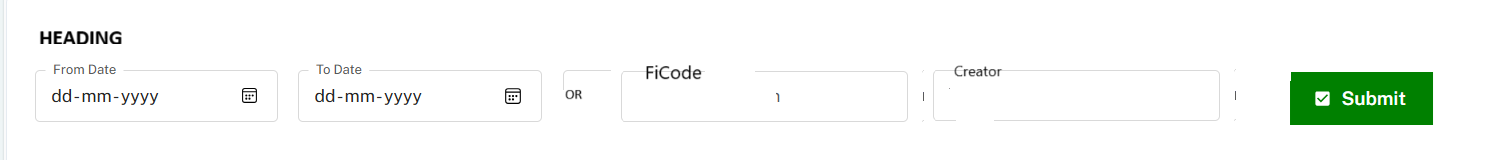
On Delete click, reload page and back to blank.

# **Post Sanction**

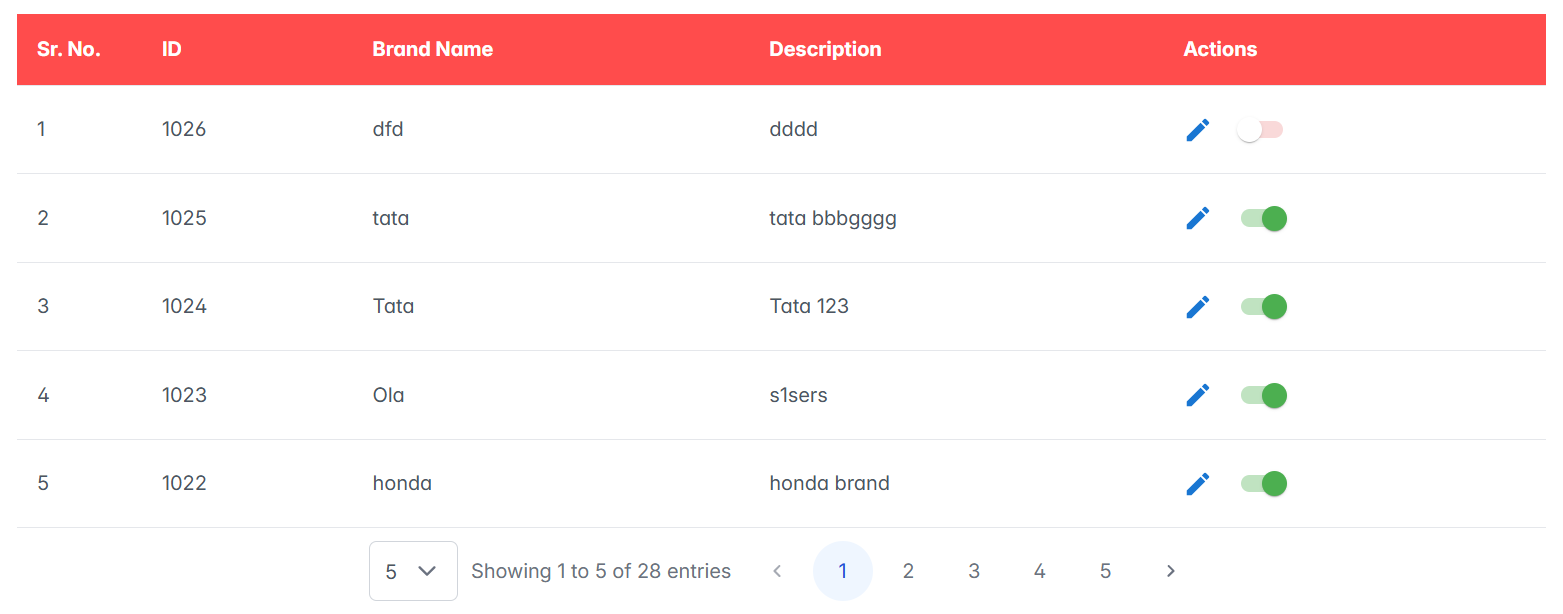
All records with **Status 6 (Ready to Push)** should be retrieved and displayed in a table. Users can filter the records based on **FI Code**, **Creator** or **Dates** to narrow down the results. The table will display the following fields for each record: **SM Code**, **FI Code**, **Creator**, **Branch**, **Group**, **Second e-Sign Status**, **Upload Document (Pronote)**, **Comments**, and an **Action button** for further processing.

If the **SM Code** is blank, a clickable option should be provided in the table to **download a One Pager** related to that record. Uploading the **Pronote Document** is mandatory, and if it has not been uploaded, the record should be restricted from proceeding to the next step until the document is uploaded. Additionally, a **Push to Audit** button should be available, which, when clicked, will update the record’s status to **7** (Audit). This ensures that only complete records with all necessary documents can progress further in the workflow.

# **Post Sanction Design**



Example



-Search by these fields

-Bind Table below with pagination, search box and export to excel feature.

-popup on document click (Table) option to add Pronote and any other document.

-Icon for Move to Audit.

# **Ready To Audit**

Retrieve all records with **Status 7** and provide a search functionality that allows users to filter based on **Date**, **Creator**, **Branch**, and **Group**. Additionally, include an option to **export the data to Excel** for easy reporting and further analysis. The UI will display the following fields for each record: **SM Code**, **Sanction Date**, **Amount**, **Schmcode**, **Account Name**, and **Brand/Model (VHCase), Details (**popup will be available that allows users to view **Person Details**, **Final Details**, and **Documents** tabs view). Users will also be able to download the related documents directly from the popup. An **Action button** will be provided; if the action is approved, the system will update the record's **status to 8**, indicating successful processing and transition to the next stage. Additionally, a **Reject button** will allow users to send the case back to the post-sanction stage, updating the status to **6** in the database. When a case is rejected, a **popup** will appear for users to add comments or remarks explaining why the case is being rejected, and these remarks will be saved in the database. Furthermore, logs for every action and status change should be inserted into the **FiJourney** table to track the case’s progress.

# **Neft Done**

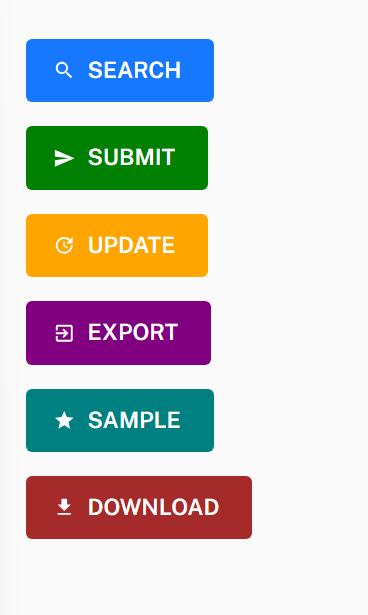
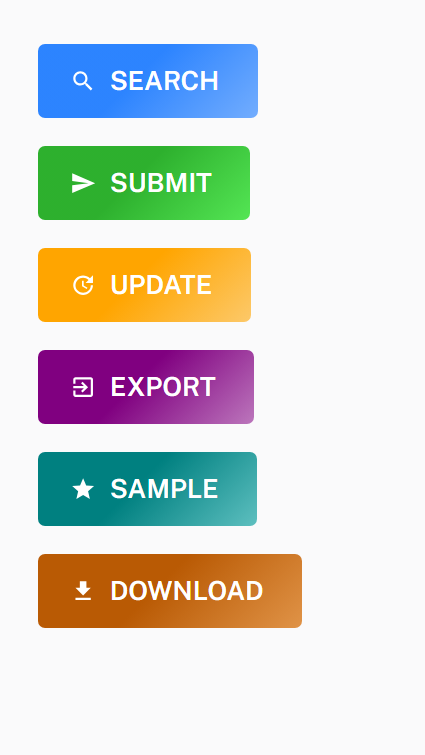
Section 1: File upload option for bulk NEFT Approved cases. Provide an option for bulk **NEFT Approved cases** to be uploaded as a file. Once the file is uploaded and the records are approved, the system will automatically update the **status to 9** for all successfully processed cases, indicating that the NEFT cases have been approved and are ready for further actions.

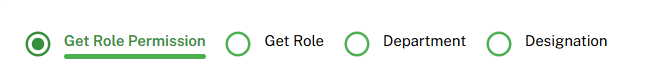
Section 2: File upload for Reject NEFT case. Allow users to upload a file containing **Rejected NEFT cases**. Upon file upload, the system will process the records and update the **status to 6** for all rejected cases, signalling that these cases need to be sent back to the post-sanction process for review or further actions.

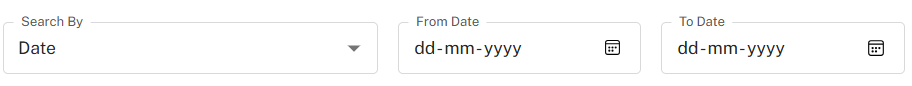
Section 3: Retrieve all records with **Status 8** and provide search functionality for filtering these records based on **Date** or a combination of **Creator**, **Branch**, and **Group**

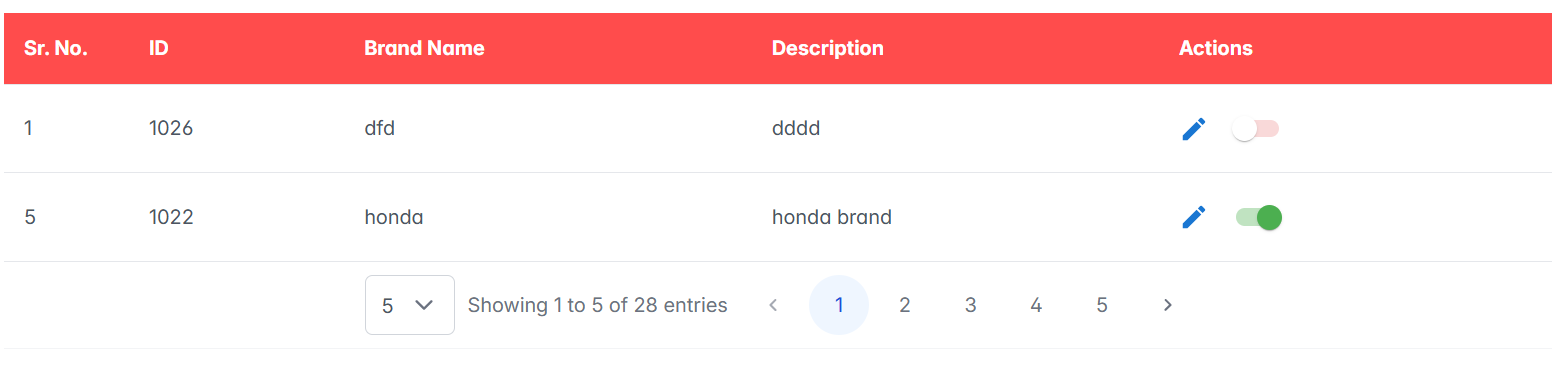
When the user selects particular FI to approve, they will then be presented with an option to select records from a table. After selecting the desired records, the user can click on the **Approve** button to proceed. If the **SM Code** of the selected FI is "PD," the system will automatically mark the status of the selected records as **Status 9** (approved or ready for NEFT). However, if the **SM Code** is anything other than "PD," then it will insert the approved records into the relevant bank-related table, ensuring proper tracking and processing by the associated bank.

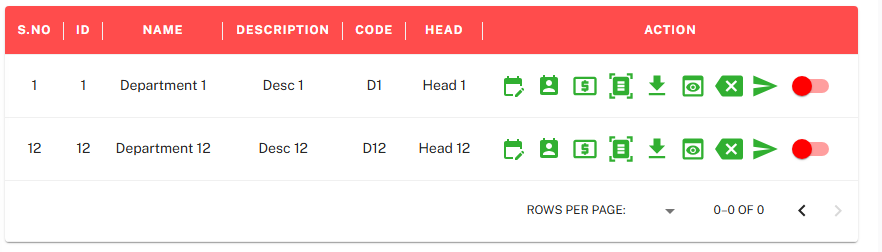
# **Common Design**









# **Master’s**

* 1. **Add Menu**

Insert / Edit / Delete (Active- deActive)  
Add three radio buttons in the UI for Main Menu, Sub Menu, and Page Name, allowing single selection at a time. For Main Menu, provide a textbox for input. When adding a Sub Menu, display a dropdown (DD) to select the associated Main Menu and a textbox for the Sub Menu name. Similarly, when adding a Page Name, provide dropdowns to select the Main Menu and Sub Menu, along with a textbox for the Page Name**. [Tables:** menu**]**

- During each insertion of a UI into the menu table, the number of entries corresponding to the users should be inserted for the respective menu ID in table [menu permission]

* 1. **Roles**

Add four radio buttons in the UI Role Permission, Role, department, Designation.

-Menu permission: there are dropdown for Submenu, Role, users – select role from dd and users bind in next dd then select submenu - all respected pages will visible with multi select checkbox. User can check the select the required Ui’s and change the permissions that update in database field IsActive true. Option for select all at the top

-Roles: Textbox for insert roles, submit. Edit- open popup with editable text box and submit button, Active, de-active icon. Show records in table with pagination

Textbox for Role, Department and dd of department and textbox in Designation section.

* 1. **End Points**

Insert End points-textbox for service name, controller name, Action name and dropdown for Page UI (save Id of URL in table). Bind all records in table with pagination, search option and Active deActive option.



# **Add** **Employee**

Section 1: Add new employee with all the input fields which are in current LOS. Use Dropdown for Role, Department, designation, reporting manager. Add all required validation then save record.  
Section 2: search box (autocomplete) for search user. Radio buttons for Education details, Last company Details, KYC, Salary Details, Emergency details, Borrowing details, Documents. Input details for these sections are same as current LOS. Save all documents in folder (. /EmployeeDocs/EmpID/Filename). Submit button for insert. View button redirects to another UI where all emp data binds in table with search option and pagination, Active/de-Active icon, Edit icon. On Edit click redirect back to the main page with autofill current details in editable mode.

# **Add Creator**

Insert / Edit / Delete (Active- deActive)  
[AddCreator - Paisalo](https://predeptest.paisalo.in:8083/LOS/Master/AddCreator)  
 Include the same fields currently used in the LOS.  
 Submit and View button. View button redirects to another ui where all records bind in table with search box and pagination, edit icon. On edit redirect to main page with all current details in editable mode. Update record.  
Validation: Duplication not allowed, with combination of creator + tag. Only Single entry will insert with combination of creator + tag.

# **Add Branch**

Insert / Edit / Delete (Active- deActive)  
[AddGroupMaster - Paisalo](https://predeptest.paisalo.in:8083/LOS/Master/AddGroupMaster)  
Include the same fields currently used in the LOS.  
 Submit and View button. View button redirects to another ui where all records bind in table with search box and pagination, edit icon. On edit redirect to main page with all current details in editable mode. Update record.  
-       Remove {Code} column from this form , get max code value from table add +1 then save

# **Add Group**

[JlgGroupMaster - Paisalo](https://predeptest.paisalo.in:8083/los/FI/JlgGroupMaster)  
For the Edit functionality, allow users to select the Creator, Branch, and Group Code. The remaining fields should auto-fill based on the selection, but users should be able to edit the details. For the Insert New functionality, allow users to select the Creator and Branch, then enter the other required values, including Group Name, Village Name, Centre Name, CSO Code, BM Code, Pre GRT, and GRT. The system should automatically fetch the maximum Group ID and add 1, based on the combination of Creator and Branch.

# **Vehicle Master**

Insert / Edit / Delete (Active- deActive)  
 Add a radio button for Fuel, EV Brand, EV Model, and Vehicle each of these four types, allowing users to select one at a time. Only the fields relevant to the selected type should be visible. Submit details and visible in table (pagination and search option). Edit and Active deActive option in able. When click on edit popup open in editable mode then submit.

# **Add Banner**

Insert and Delete Banners for mobile use  
 Radio button for Type of banners  
 SWAP Message – ddl for type, textbox Title, 4 banner upload option, editor for description then submit. Bind record in table (pagination and search)  
 Banner & Advertisement – ddl for type, textbox for advertisement, description, banner upload option then submit. Bind record in table (pagination and search)  
 Flash Message - ddl for type, textbox heading, banner upload option, editor for description then submit. Bind record in table (pagination and search)

Old Url: <https://predeptest.paisalo.in:8083/los/Notification/BannerPosting>

# **Branch**

# **GUARANTOR**

Radio Button for Add Guarantor and Guarantor Crif

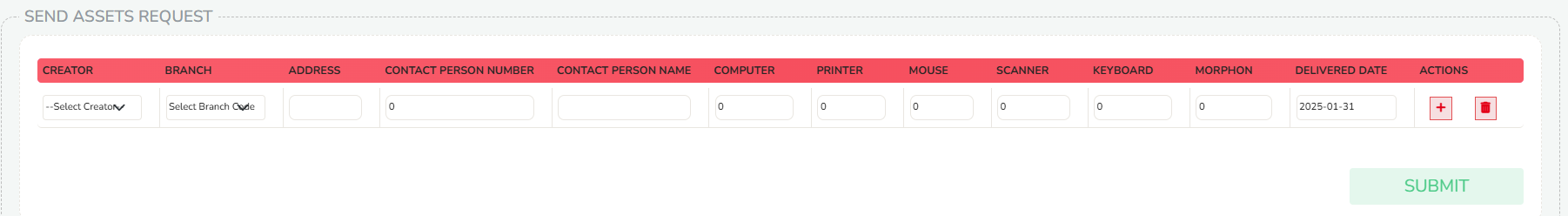
Add Guarantor: Insert fields Creator, Ficode, name, guardian name, date of birth, gender, relationship with the borrower, region, caste, Aadhaar (mandatory), PAN or voter ID (only one required), mobile number, and address (Address 1 and 2 are required, Address 3 is optional), ddl for stats, pin , file uploader for profile pic. Then submit.

Generate Crif: get all Input fields as same in current los after submit, show details in Popup.

[FiGroup - Paisalo](https://predeptest.paisalo.in:8083/los/Fi/FiGuaranter)

# **Asset**

**Asset Request**: In the first section, create an asset request form with fields identical to the ones specified below. Include an option to "Create Multiple Requests" by clicking a button. The data entered should be saved as an array when the submit button is clicked. The saved records will be displayed in a table below. Any records closed by the accepted person will also be visible in this table. In the "Action" column, users can add received remarks and upload files related to that specific record.



**Asset Approval**: There should be a search feature to filter records by Creator and Branch. All filtered records will then be displayed in a table. An "Action" button will be available for approval purposes.

**Asset Dispatched:** All EVP-approved records will be shown in a table, with an option for users to add remarks. Filters (All, 1st Approval, 2nd Approval, Rejected and dates) additionally, users can either add remarks or upload an image and close the request.