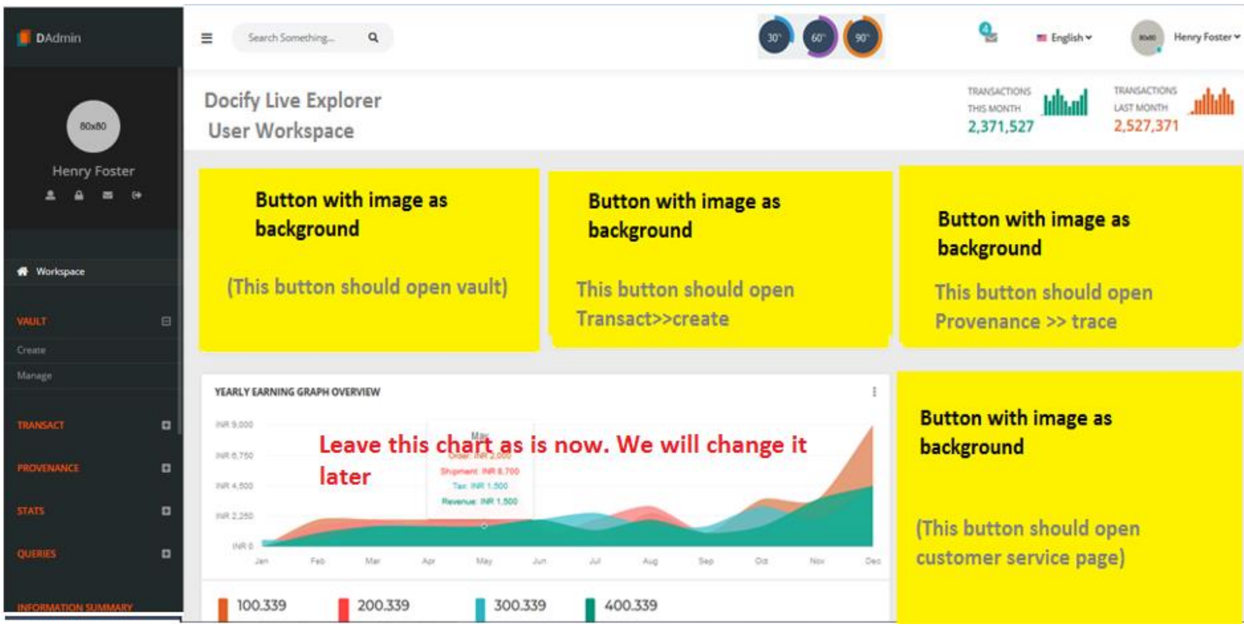
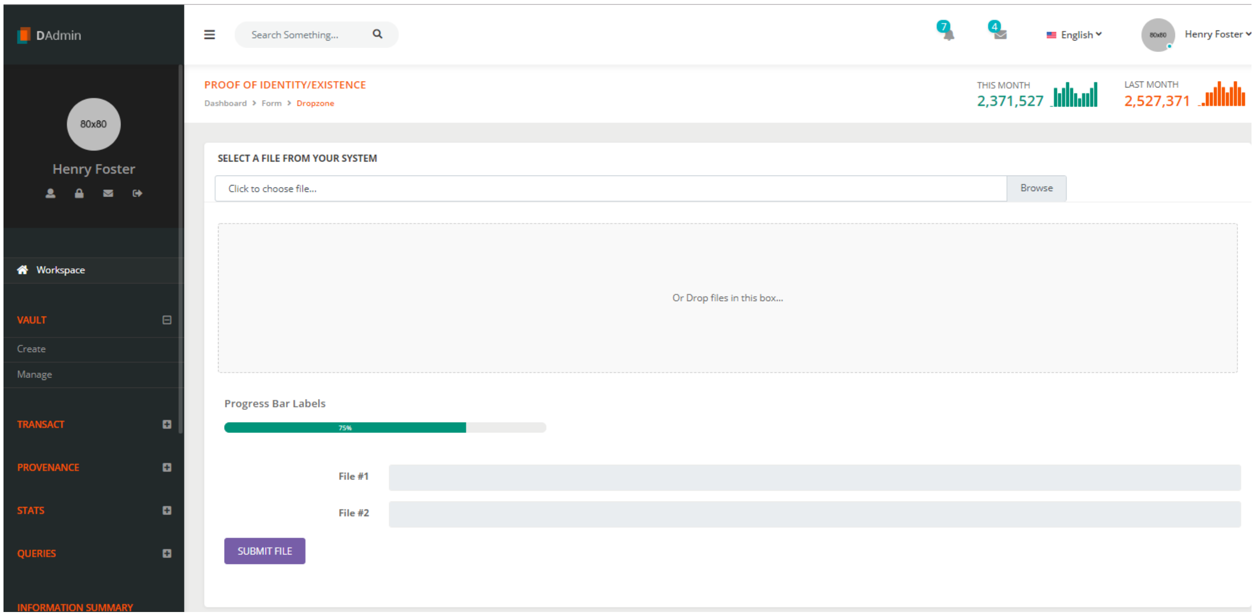



This is the main workspace

Important Notes:

1) The Yellow rectangles are button that will have an image on them. This will be clickable buttons that will navigate the users to various places on the application(see text description).
2) Leave the other portions of workspace as is. We will need changes to it but they are low priority as of now. We can make those changes once everything else has started working.
3) Please note the small icons on top status bar with a number inside them. These icons will have usage1(space used in MBs), usage2(space remaining in MBs), days(number of days of service remaining)
4) The search box that is present on left side on top status bar can be removed as of now.



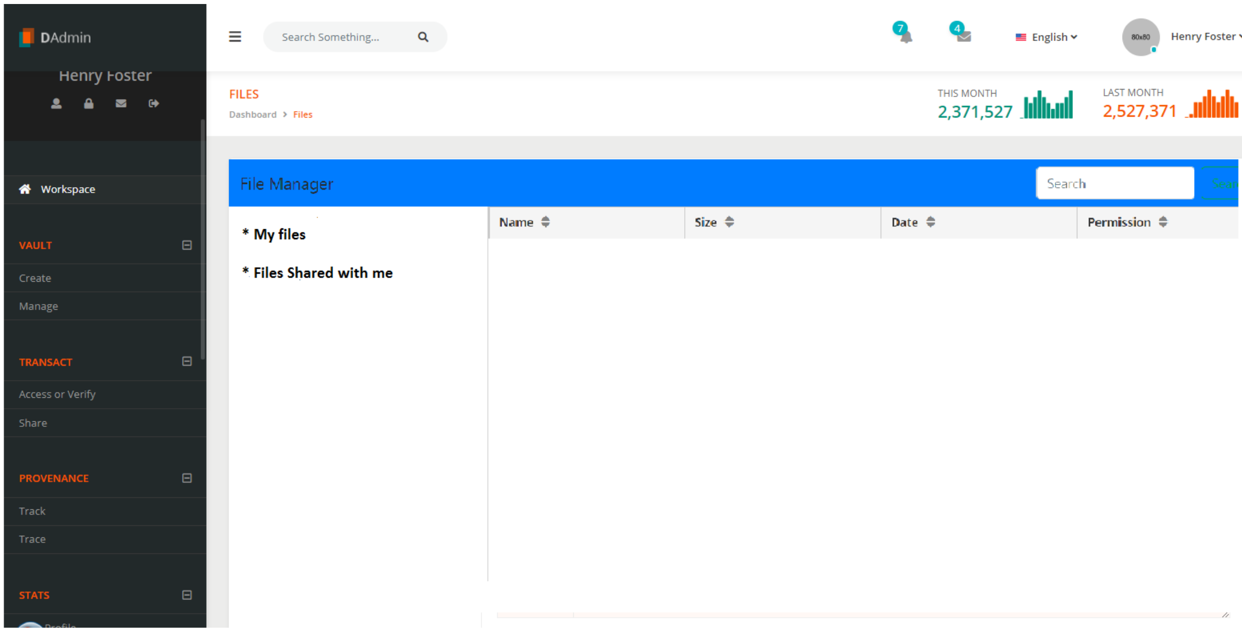

## Inside Vault >> Create

Step1: User will select a file from her system

Step2: User then clicks submit to upload the files to the file manager. The progress bar shows this progress. The file should be written to the backend mongodb database.

Step3: A modal message box appears that says “Do you want to share this file NOW?”. There should be 2 buttons “YES” and “LATER”. If the user clicks YES then user should be taken to Transact>>share.

Note: After the file Is selected, the file name should be displayed as well as a progress bar that displays its upload status. UI elements will need to be picked from the template and added to this page.

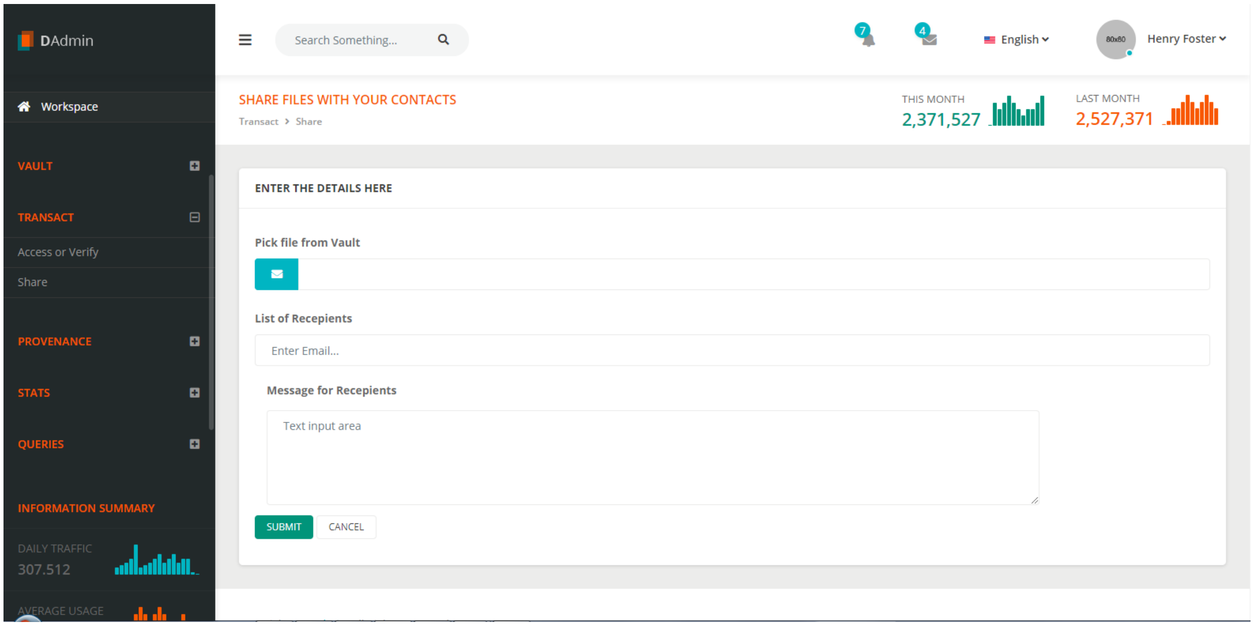



## Inside Vault >> Manage

The user will see all the files he created and the ones which others have shared with him.

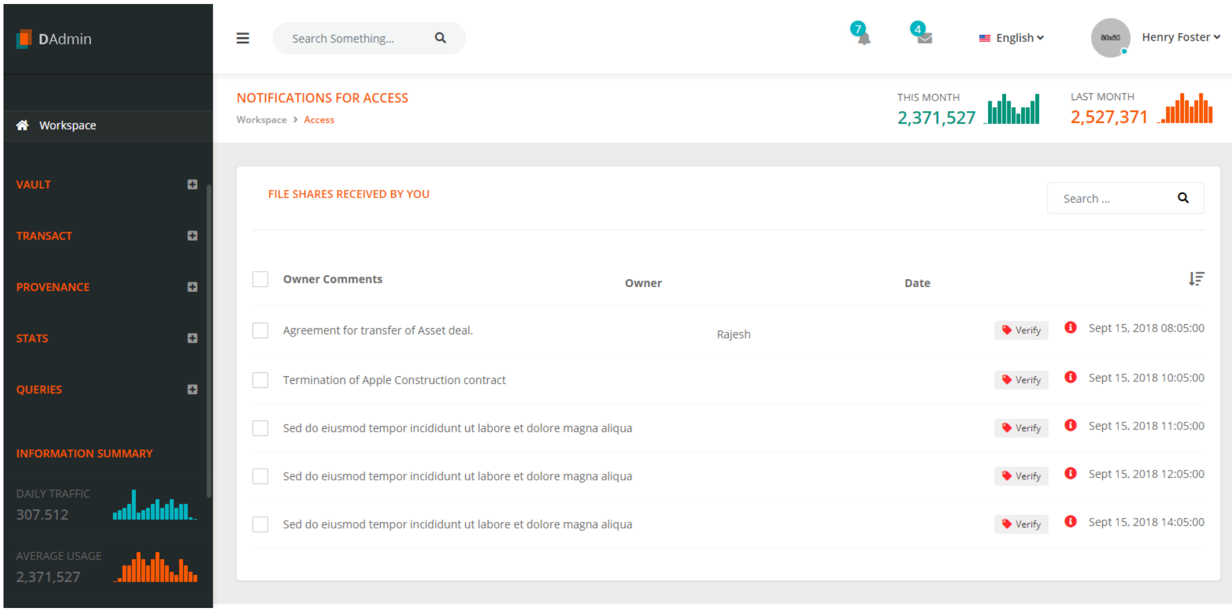
Here, user can

1) Preview the file in a modal dialog
2) Basic functionalities like Select, Rename, navigating inside directories etc.




## Inside Transact>>Share

1) On this page the user will click on the envelope icon. This will navigate the user to the file manager. On the file manager user will select a file by double clicking on it (or by clicking Share button). This will select that particular file in memory and will take user to this page.
2) The selected file name will appear in front of the envelope icon
3) User then adds his contacts who he want to share this particular file with. The email address text box is AUTOCOMPLETE. The user should get a matching contact as per what he types.
4) The user then enters a comment inside of Text Input area.
5) The user clicks on submit.
6) Submit initiates 2 actions
    a. It will send an email to the recepients that this file has been shared with them. The email will have the link of this file.
    b. A method on Docify server will be called to add these users to the file record(I will share the details of this api later).
7) We need to add a progress bar that will be 100% once both a and b are complete.



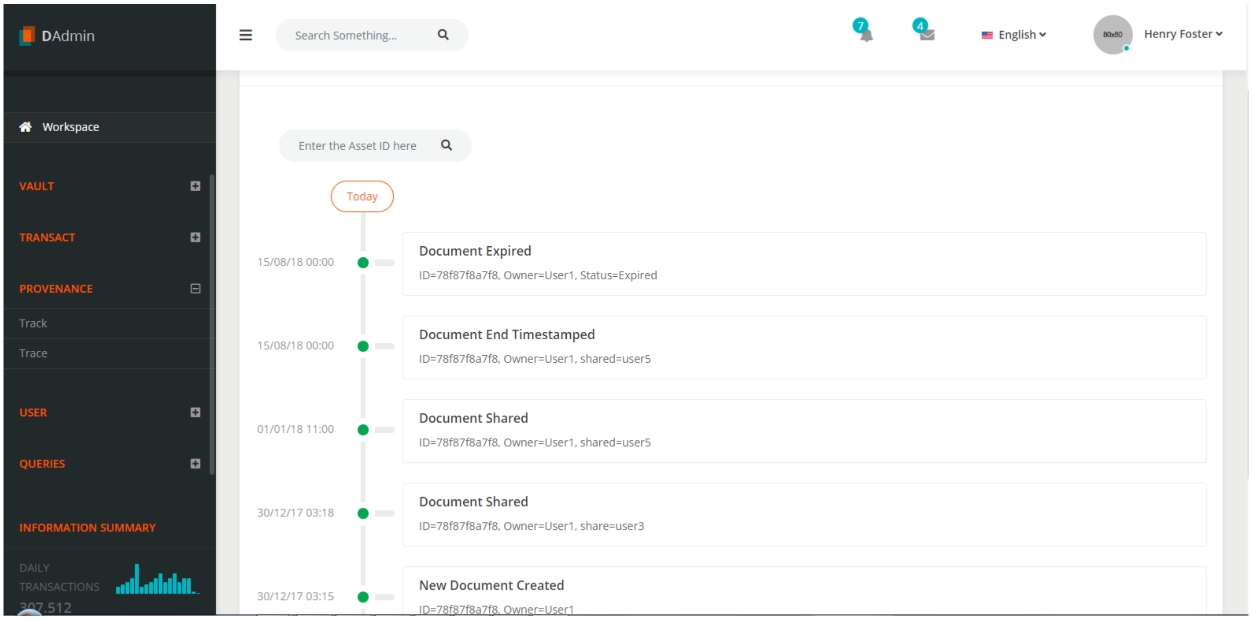

## Inside Transact >> Access

On this page user will receive notifications from users for the files other users have shared with this user.

1) Each line item is like a hyperlink. When the user clicks anywhere on the description text. The user should be taken to the file explorer window where this file gets opened in a preview mode.
2) When the user clicks on the tag “Verify” the user should be taken to the “Provenance>>Trace” page with the details of this document opened.
3) After a user has viewed a file, the checkbox should be checked against that particular line.

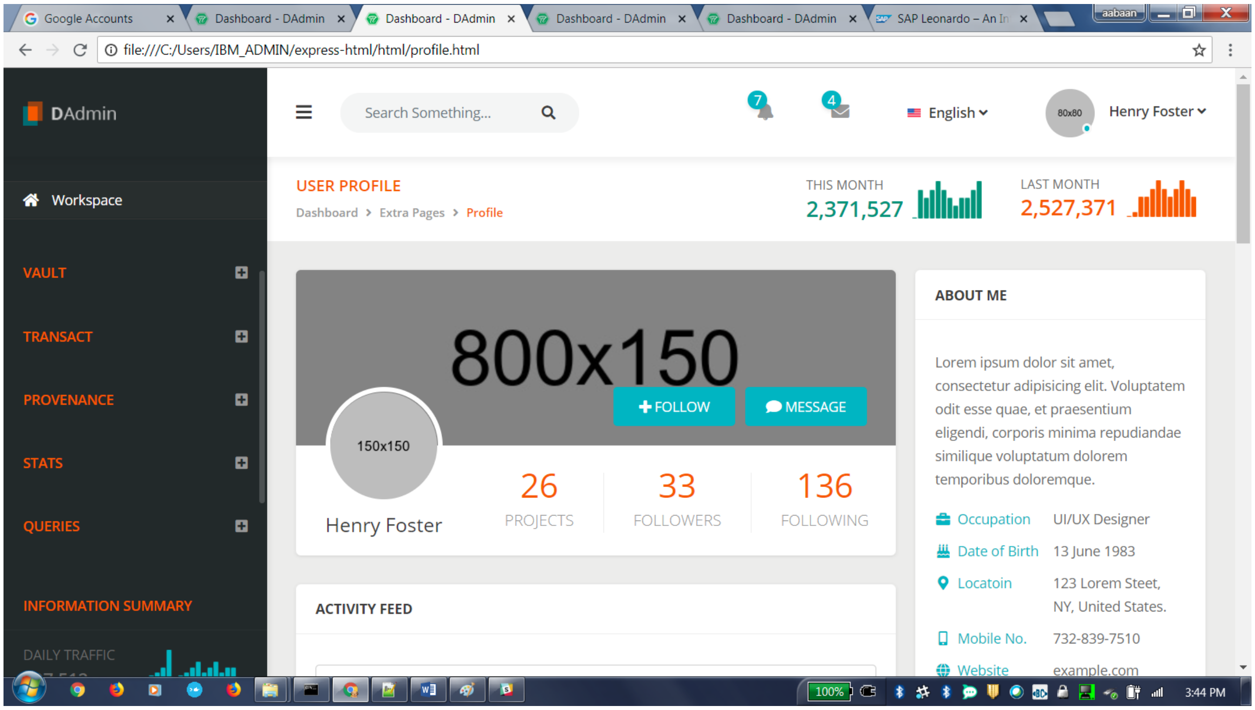
Inside Asset>>Track

This page is under design. Will update later. Leave it for now.




Inside Provenance >> Trace

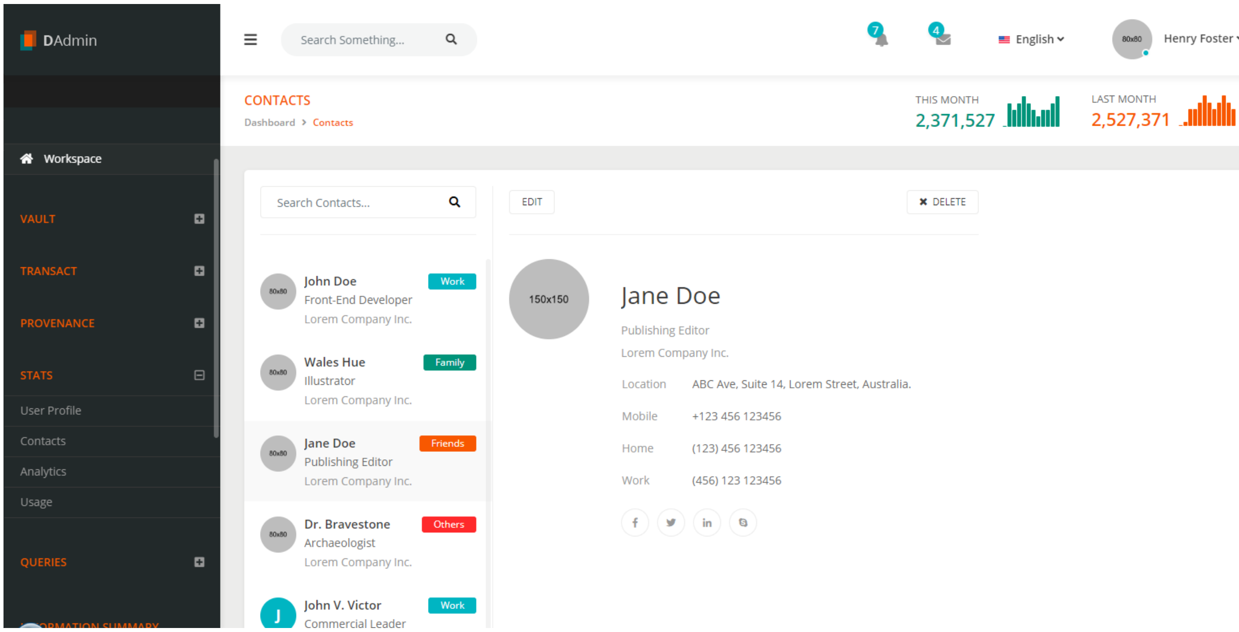
1) The user can view this page to see the history of a document. There will be a list of document related events displayed (in reverse chronological order).
2) Each of these list items are clickable. When any of them are clicked a pop up window displays that details of that particular transaction. This detail will be returned through an API call by the Docify backend server2.




This is the user profile page

User can navigate to this page from sidebar or user Icon on top menu

1) Populate all the fields for which we have information.
2) Make available a set of 10-12 Avatars that people can use in case they don't wish to upload their pictures. These avatars should be picked automatically by the application when a user signs up. The avatars should be displayed in all the places where there should be user image.

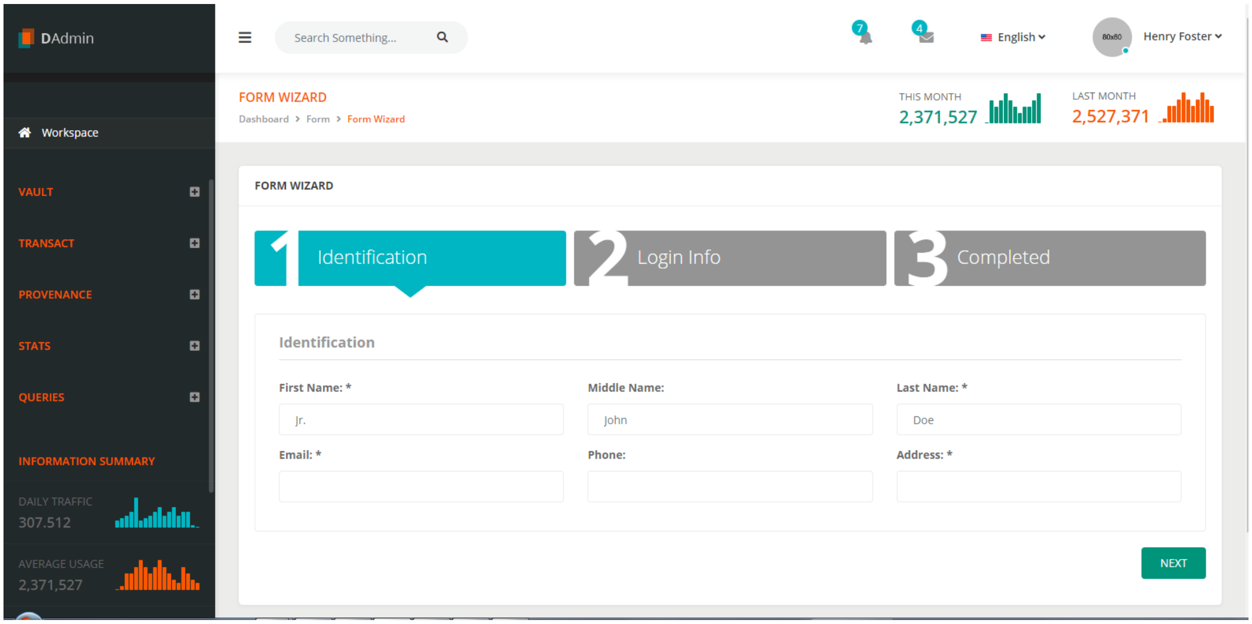



## Inside Stats >> Contacts

This page shows the list of existing contacts this user has saved.

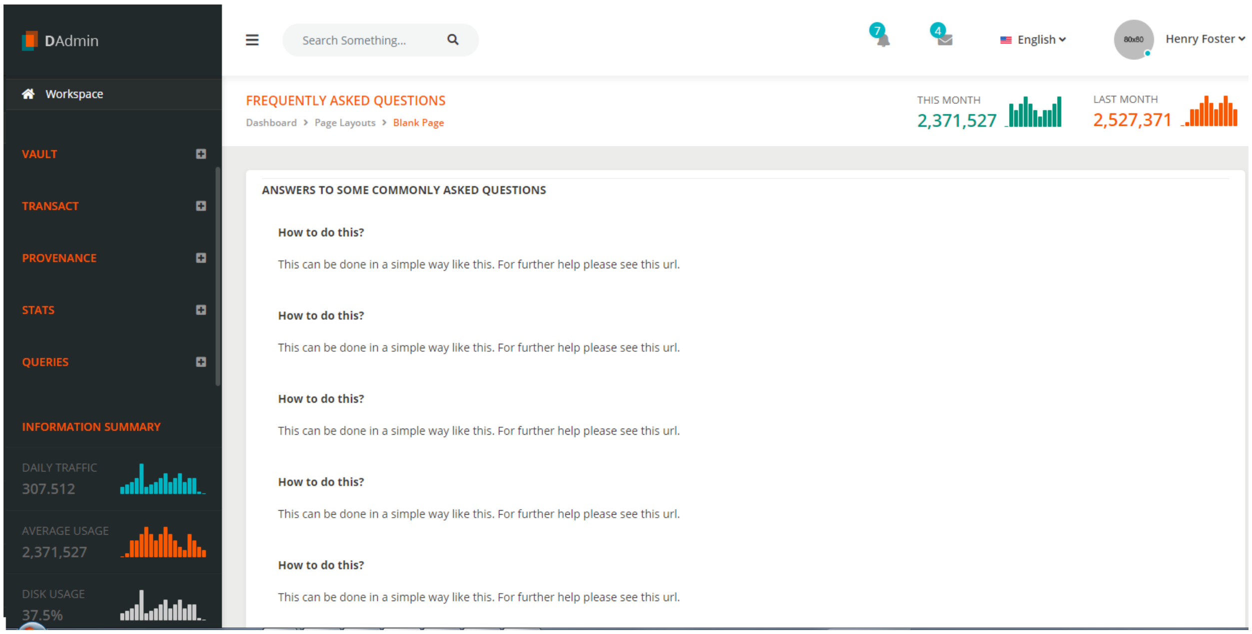
User can create contacts by the following ways:

1) Either creating and saving contacts on this page directly. In this case full record of a contact will be created.
2) Or creating a new contact when using “Transact>>Share”. In this case only emailID of the user will be saved in contacts.
3) The page should give the user an option to “Edit” or “Delete” an existing contact.




## Inside Queries>>Support Centre

1) User can use this page to raise tickets if he/she is facing any problem with the application
2) Reuse the elements on this page to record user name, email, phone, and the problem description. Once the user submits he should get the ticket number displayed on the screen. You will need to throw out the unneeded things and keep only relevant elements.




## Inside Support>>FAQ

1) The questions and answers are all shown as labels. However, we need the ability to load these questions and answers dynamically from a json file like below:

Faq.json (an array of questions and answers which we can later edit)

[

{“question”:”how to share a file”,”answer”:”a file can be shared by navigating to the left hand side navigation sidebar Transact>>Share”},

{“question”:”how to view user profile”,”answer”:”click the small icon under the user photo on the left hand side navigation sidebar”},

…..

]

# SideBar

The user image space should display the avatar.

Sidebar should remain the same on the pages. The 3 bar charts on the sidebar can be populated with user information at a later point in time.