

Suvit
(Accounting Automation Software)

By
Princy Gajera (ID No. 19CEUOS157)

**A project submitted In
partial fulfillment of the requirements
for the degree of
BACHELOR OF TECHNOLOGY
In Computer Engineering**

Internal Guide
*Prof. Jatayu Baxi,
Assistant Professor,
CE Department,
Dharmsinh Desai University,
Nadiad.*

External Guide
*Mr.Kalpesh Zalavadiya,
Project Manager,
Suvit Fintech.*



**Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2023**

CERTIFICATE

This is to certify that the project work titled

**Suvit
(Accounting Automation Software)**

is the bonafide work of

Princy Gajera (19CEUOS157)

**carried out in the partial fulfillment of the degree of
Bachelor of Technology in Computer Engineering at
Dharmsinh Desai University
in the academic session
December 2022 to April 2023.**

*Prof. Jatayu Baxi,
Assistant Professor,
CE Department,
Dharmsinh Desai University,
Nadiad.*

*Dr. C. K. Bhensdadia
Head,
CE Department,
Dharmsinh Desai University,
Nadiad.*



**Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2023**



INTERNSHIP CERTIFICATE

Date: 28th March 2023

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Princy Gajera** Student of **Bachelor of Technology (Computer Technology)** **Dharmsinh Desai University** is currently pursuing her internship from **12th December 2022 to 14th April 2023** at **Suvit Fintech Pvt. Ltd., Surat.**

She is working on a project titled **Suvit (Accounting automation software)**. This project is aimed at a college Internship. As part of the project, she is working as a **SDE-Intern**.

During her Internship, she has been exposed to a variety of different processes and has essentially been taken on a rollercoaster ride where she had a myriad of enriching adventures in our innovative software world and yet she managed to demonstrate her inclination to learn new skills swiftly and therefore her performance exceeded our expectations.

She is able to complete the project on time while being diligent, hardworking, and inquisitive. We wish her all the best for her life and career.

With Regards,



Mr. Ankit Virani

Suvit Fintech Pvt. Ltd.

Suvit Fintech Private Limited

901, Infinity Tower, Near Ayurvedic College, Station Road, Surat-395003, Gujarat, India.

📞 +91-9558261955

✉️ admin@suvit.io

🌐 www.suvit.io

ACKNOWLEDGEMENT

It is indeed a great pleasure to express my thanks and gratitude to all those who helped me during this project. This project would not have materialized without the help of many people who asked me good questions and rescued me from various red tape crises.

Theoretical knowledge is of no importance if one doesn't know the way of its implementation. I am thankful to my University and Suvit to provide me with an opportunity to apply my theoretical knowledge through the project. I feel obliged in submitting this project as a part of my Internship.

I would like to take this opportunity to express my humble gratitude to my external project guide **Mr. Kalpesh Zalavadiya**, and internal guide **Prof. Jatayu Baxi**, under whom I undertook my project. Their constant guidance and willingness to share their vast knowledge made me enhance my knowledge and helped me to complete the assigned tasks to perfection. Without their effort, support, and astonishing testing ability this project may not have succeeded.

With Sincere Regards,
Princy Gajera

TABLE OF CONTENTS

Chapter	Page
I. Introduction	1
II. About the System	3
III. Implementation	7
1. Document Processing	7
2. User Management	13
3. Notifications	23
4. HubSpot	28
5. Split Company	30
6. Intercom	35
IV. Conclusion	39
Bibliography	40

LIST OF FIGURES

Figure	Page
Chapter II	
Fig. 2.1.1 Use Case Diagram	3
Fig. 2.2.1 Signup Flowchart	4
Fig. 2.2.2 Document Management Flowchart	4
Fig. 2.3.1 Home screen	5
Chapter III	
Fig. 3.1.1 File Upload Screen	8
Fig. 3.1.2 User's File	9
Fig. 3.1.3 Processed File	9
Fig. 3.2.1 Create Admin	13
Fig. 3.2.2 Admin Flow-Chart	14
Fig. 3.2.3 User Flow-Chart	14
Fig. 3.2.4 Users Register Successfully	15
Fig. 3.2.5 Mail Sent Log	15
Fig. 3.2.6 Mail Sent to New Registered User	15
Fig. 3.2.7 Email already exists	16
Fig. 3.2.8 User Login Successfully	16
Fig. 3.2.9 Email Not Exists	17
Fig. 3.2.10 All users register as per the excel	17
Fig. 3.2.11 Auto-generated password stored after encryption	18
Fig. 3.2.12 Mail sent with an auto-generated password	18
Fig. 3.2.13 Error generated in one of the users from excel	19
Fig. 3.2.14 Generated excel with error logs	19
Fig. 3.2.15 Error generated in one of the users from excel	20
Fig. 3.2.16 Generated excel with error logs	20
Fig. 3.2.17 Create category	21
Fig. 3.2.18 Create a Product under the Category	21

Fig. 3.2.19 All Products under the Phone Category show up	22
Fig. 3.2.20 All Products under the Laptop Category show up	22
Fig. 3.3.1 Notifications Flowchart	23
Fig. 3.3.2 Different templates to send mail	24
Fig. 3.3.3 Select all clients to whom the user wants to send mail	24
Fig. 3.3.4 Mail Preview	25
Fig. 3.3.5 Mail sent confirmation	25
Fig. 3.3.6 Mail receive by client	25
Fig. 3.3.7 Different templates to send notification	26
Fig. 3.3.8 Select notification template	26
Fig. 3.3.9 Select all clients to whom the user wants to send a notification	27
Fig. 3.3.10 Notification sent confirmation	27
Fig. 3.4.1 User Register to the Suvit	28
Fig. 3.4.2 User details added to the HubSpot Contacts list	28
Fig. 3.4.3 User purchase plan and become Active Client	29
Fig. 3.4.4 Clients details added to the HubSpot Company list	29
Fig. 3.5.1 Activity Diagram for split company	30
Fig. 3.5.2 List of all Companies in Electron	31
Fig. 3.5.3 Subscribe company to Suvit	31
Fig. 3.5.4 Split company from Tally	32
Fig. 3.5.5 Two new companies added with same name &different dates	32
Fig. 3.5.6 All three companies shows in Suvit	33
Fig. 3.5.7 Parent company enabled	33
Fig. 3.5.8 One child company enabled	34
Fig. 3.6.1 Intercom implemented features	35
Fig. 3.6.2 Messenger without Registration	36
Fig. 3.6.3 Welcome Post to Registered User	36
Fig. 3.6.4 Chats Integration	37
Fig. 3.6.5 Message receive in intercom inbox	37
Fig. 3.6.6 Custom Bots Integration	38
Fig. 3.6.7 Checklists Integration	38

Chapter I

Introduction

Suvit is a universal platform for the integration of various accounting and tax programs into a single office environment. Suvit helps you stitch together a seamless journey in managing financial data and operations through a single platform -whether you are a taxpayer or a tax expert, it does not matter.

The main task is to provide a convenient open architecture for processing huge financial data flows and automating finance management processes. In suvit, we can automate Banking, Payment, Receipt, Purchase, and Sales.

Suvit aims to become a smart digital platform built to bridge the gap between taxation, accounting, banking, auditing, professionals, and users involved in such sectors to shorten the data life cycle for a better response.

Suvit uses an architecture that facilitates on boarding the taxpayers and tax experts of the country and integration with government portals. This enables swift movement of data, transparency in operations, and ease of creating a centralized repository. The solution results in massive time savings (to the tune of 80%) and an increase in process efficiencies.

Suvit offers an accounting automation solution that will reduce all your clerical and manual tasks without any hurdles.

What Is Accounting Automation?

It is a process of automating day-to-day clerical and repetitive accounting tasks such as data entries, reconciliation, voucher entries, tax filing, etc with the help of advanced technologies like Artificial Intelligence and Machine Learning.

Third-party software or tools performing all these tasks is known as accounting automation software.

How accounting automation can help accounting team collectively?

- Better Communication:

An accounting automation tool can be your one-stop solution for all communication with its ability to in-app chat system. Send and receive documents from clients and team members on a single platform.

- Better Collaboration:

Share and manage documents on a single platform. You can also manage access to documents and other features. Having everything in a single accounting automation platform will eliminate the process of waiting for documents and approvals.

- Opportunity To Work On Futuristic Approaches:

Apart from saving time, cost, and effort, accounting automation can actually spare you time and provide analysis to work on financial planning, analyzing and tracking all the past activities and data. So that the whole team can actually focus on future business plans and strategies. Automation increases the productivity and efficiency of your whole accounting team and significantly decreases the business running cost.

Chapter II

About the System

2.1 Use Case Diagram

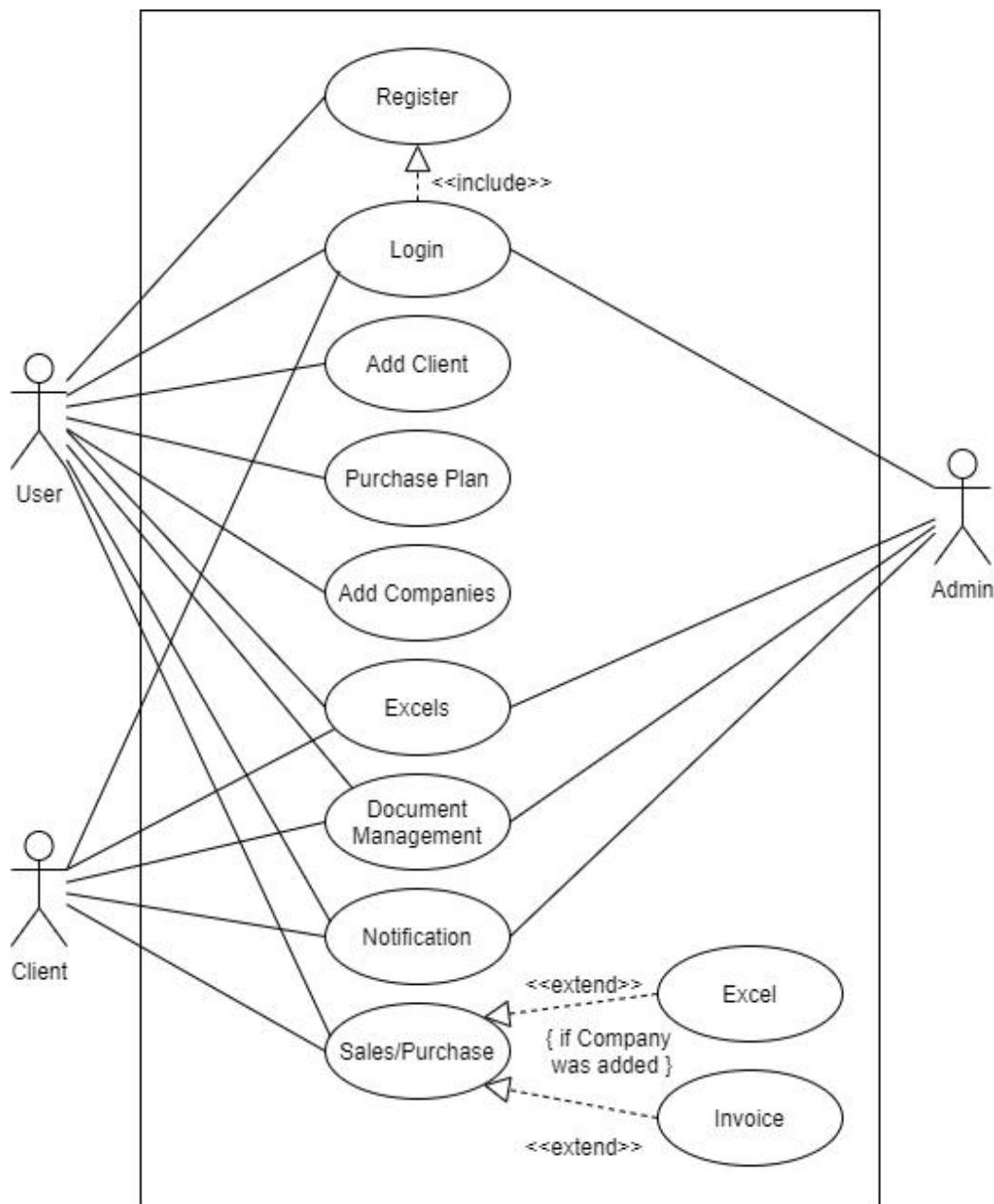


Fig. 2.1.1 Use Case Diagram

2.2 Flow charts

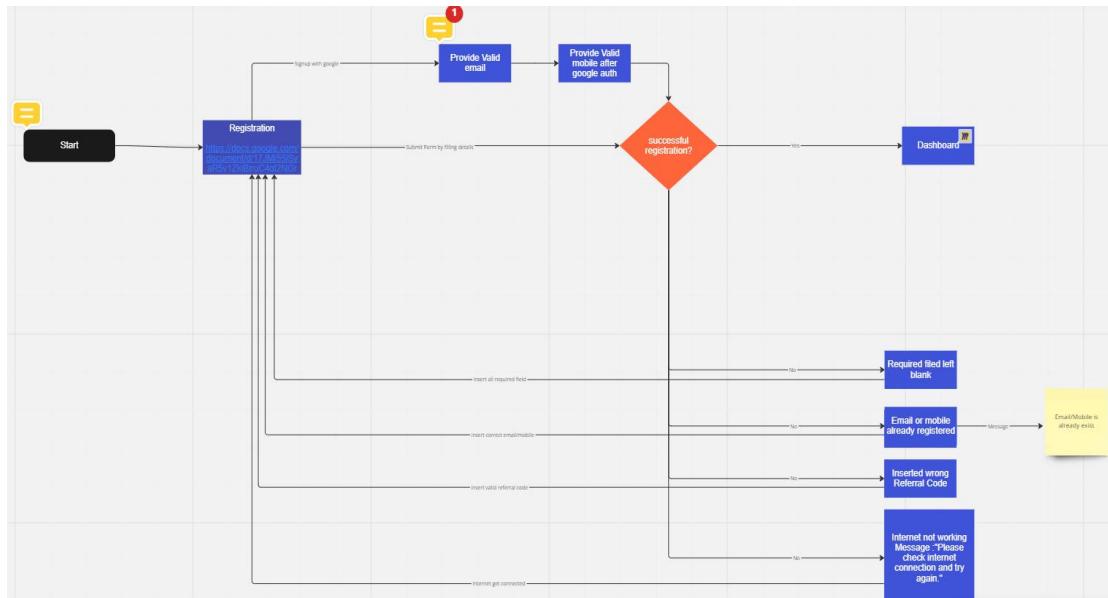


Fig. 2.2.1 Signup Flowchart

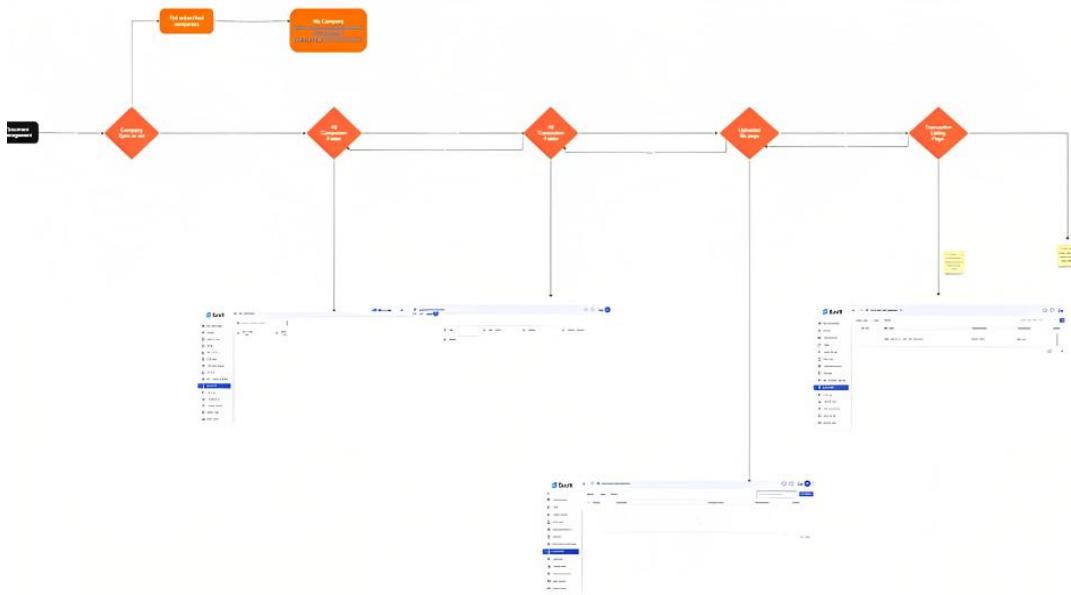


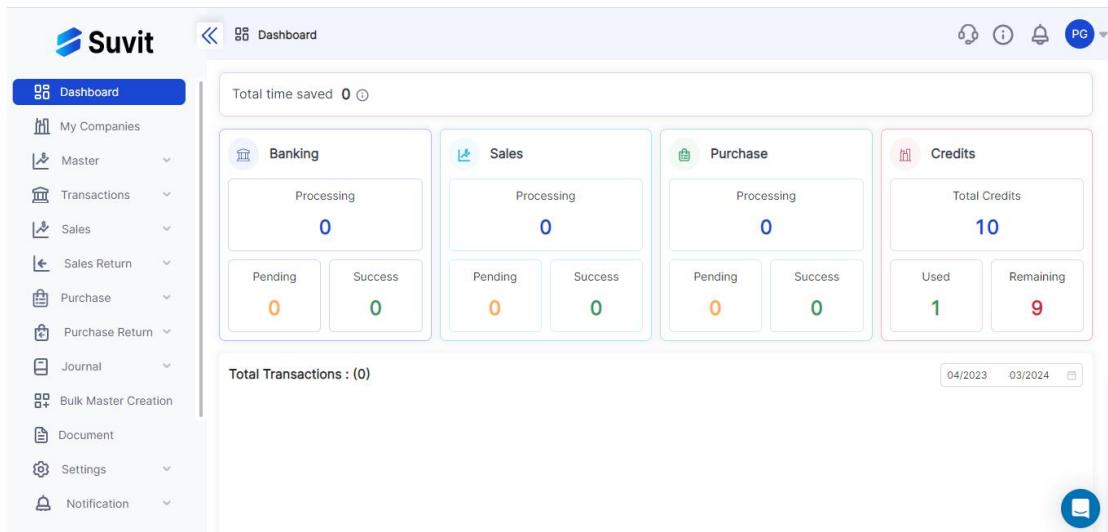
Fig. 2.2.2 Document Management Flowchart

2.3 Brief introduction about all modules of Suvit

- **Sign In**

Two methods the user can log in.

- Providing a registered email and password.
- Sign up with Google.



2.3.1 Home Screen

- **Dashboard**

After successful login, the user will be redirected to the dashboard where he will see the on boarding screen for the account set-up.

- **My Companies**

In this module, you will be able to see the list of companies that are synced or added from Tally.

- **Master**

Ledger/ Item: Here, users will be able to see the ledger synced from Tally or created from Suvit.

- **Transaction**

In Banking Module, the user can upload the bank statement and sync it to Tally.

- **Sales/Purchase**

By using this module from Suvit, users can upload the sale, sale Return, Purchase, and Purchase Return files and sync them to Tally.

- **Journal**

By using the journal module, the user can upload the journal entry to Suvit and sync it to Tally.

- **Document**

Document Management will help the user to upload client-wise documents. Here, users will be able to see all client's folders in which they can upload and manage the documents.

This feature is introduced to speed up the process of uploading and managing all documents in a single place.

- **Settings**

There are five modules in the settings menu

My Profile: From here, users can update their profile.

My Plan: From here, users can check their credits, add-on user, or purchase a new plan. Here all current plan details are mentioned.

User Management: From here, you can add the secondary user. Also can edit, view, and delete users.

Role Management: From here, the user can create a new role and assign permission.

Client Management: This can be used to add clients by single entry or in bulk with Excel upload.

- **Notifications**

Users can send notifications to their clients regarding any updates or information that he wants to convey to their clients. Notifications can be sent, via Push Notification or Via Email Notification.

- **Tally Connector**

The Suvit desktop tally connector application will be a bridge between Suvit & Tally in your system. You can install it from your Suvit login.

Chapter III

Implementation

Tools /Technologies Used

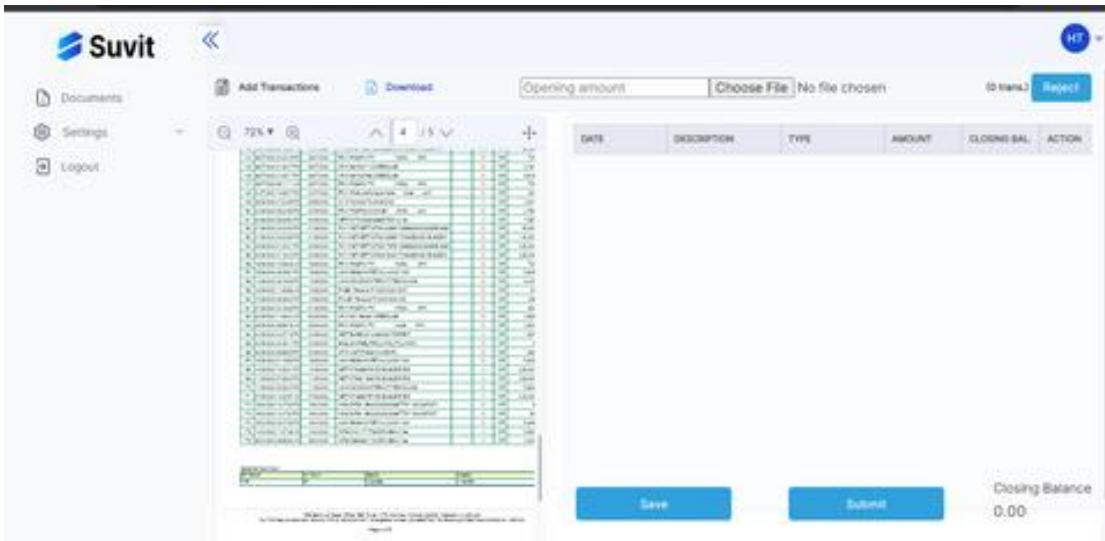
- Technologies
 - NodeJs, ReactJs, Express, Git, ExcelJs
- Tools
 - GitLab, Vs Code
- Database
 - MongoDB

Task 1

Document Processing

In Suvit, Users can upload their data sheets in any format like an excel, PDF, or scanned PDF. With the help of advanced data capture methods and machine learning, you can directly do entries in tally. No need to update every transaction manually, Accounting automation software will directly update the books/ledgers if needed. If a user uploads a PDF or scanned PDF, then it automatically converts into an excel sheet with proper formats and headers.

After this, all the files go to the admin, check the data format and validate data. Sometimes users upload files with the wrong data format or some have unnecessary data. For the further process of those sheets, we need to clean up those files, extract all the necessary fields from them, and create a new sheet.



3.1.1 File upload screen

For cleaning up those sheets, there are different scenarios for the date, narration, credit, and debit fields.

- If the Date, Narration, and credit or debit columns are there, then we can continue with that record.
- If the Date and Narration are there, but credit or debit is not there, skip that entry.
- If both credit and debit amounts are entered, then skip that record.
- If only the date column is not available then add the upper transaction's date to this one and consider that record.
- Same as if only the narration field is missing, then add xxx to that field and consider that record.
- If only the date field is there skip that entry.
- same as if only the narration field is added then skip that entry.
- If only the credit or debit column is there, then take the date from the upper transaction and add xxx as a narration and add that entry.

In the below files, the user has given so much additional information into the sheet like company name, address, IFSC code, etc. After processing sheet has only four required fields with default column title names. All the irrelevant data and columns are removed from it.

A	B	C	D	E	F	G	H
1 Account N	VADISHNAVI TECHNICAL SERVICES	Mr. VINOD RAM NAIK					
2 Address		ARAN,NIGDI PUNE 44	PUNE				
3							
4							
5							
6 Date							
7 Account N							
8 Account D							
9 Branch							
10 Drawing P							
11 Interest R							
12 MOD Bala							
13 CIF No.							
14 IFS Code							
15 MICR Code							
16 Balance o							
17 Start Date							
18 End Date							
19 Txn Date		/C Branch Co	Debit	Credit	Balance		
20 06-Jan-23		5076		30,96,589.00	41,14,048.75		
21 06-Jan-23	06-Jan-23	TO TRAN YSBI16610	99922	388		41,13,660.75	
22 12-Jan-23	12-Jan-23	TO TRAN NEFT INB:	99922	6,815.00		41,06,845.75	
23 12-Jan-23	12-Jan-23	TO TRAN NEFT INB:	99922	2,832.00		41,04,013.75	

3.1.2 User's file

A1	date	description	type	amount
1 date	06-01-2023	BY CLEARING / CHEQUE-HDF	411 credit	30,96,589.00
2	06-01-2023	TO TRANSFER-INB BSNL Payments	debit	388
3	12-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	6,815.00
4	12-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	2,832.00
5	12-01-2023	TO TRANSFER-INB--	debit	6,274.00
6	13-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	3,000.00
7	13-01-2023	TO TRANSFER-INB RTGS UTR NO: !	debit	7,50,000.00
8	14-01-2023	TO TRANSFER-INB RTGS UTR NO: !	debit	10,00,000.00
9	15-01-2023	TO TRANSFER-INB--	debit	70,194.00
10	15-01-2023	TO TRANSFER-INB 002150123211!	debit	3,30,840.00
11	15-01-2023	TO TRANSFER-INB 0021501232118	debit	6,161.00
12	15-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	20,000.00
13	15-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	22,930.00
14	15-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	17,730.00
15	15-01-2023	TO TRANSFER-INB NEFT UTR NO: !	debit	7,440.00

3.1.3 Processed file

Testing:

Row Number	Reason	Add/Skip
2,5,6	Date Modified/Added	Add
4,10,11,13,14	Insufficient Data	Skip
7,8	Narration Added	Add
15,16,19	Both Credit & Debit Field	Skip
17,18	Date & Narration Added	Add

A	B	C	D
1 Value Date	Transaction Remarks	Withdrawal Amount (INR)	Deposit Amount (INR)
2 44925.63013	EBA/Int NSEMTF2020	13.02	
3 43923	MMT/IMPS/00931810666	0 23000 . 0	
4			0
5	EBA/Int NSEMTF2020041	13.82	0
6	EBA/Int NSEMTF2020041	0 13.8 2	
7 06/04/2020		13.82	0
8 06/04/2020		0	13.82
9 07/04/2020	EBA/Int NSEMTF2020041	13.82	0
10			
11	CMS/000634116836/ADCAMSICICI_DSP MUTUAL FUND		
12 07/04/2020	EBA/EM CTD 070420/20:	-11414924	
13 07/04/2020			
14 07/04/2020	EBA/NSEMGRNPIPO/20200407203312		
15 07/04/2020	EBA/EM CTD 070420/20:	1414924	27174.57
16	EBA/NSE N 2020067/202	27174.57	14924
17		13.82	0
18		0	3714.88
19		329.24	8107.81

3.1.4 User's file

1	date	description	type	amount
2	30-12-2022	EBA/Int NSEMTF20200331/202004021	debit	13.02
3	02-04-2020	MMT/IMPS/009318106668/Rent April	credit	23000
4	02-04-2020	EBA/Int NSEMTF20200402/202004030	debit	13.82
5	02-04-2020	EBA/Int NSEMTF20200403/202004041	credit	13.82
6	06-04-2020	XXX	debit	13.82
7	06-04-2020	XXX	credit	13.82
8	07-04-2020	EBA/Int NSEMTF20200406/202004070	debit	13.82
9	07-04-2020	EBA/EM CTD 070420/20200407222111	credit	11414924
10	07-04-2020	XXX	debit	13.82
11	07-04-2020	XXX	credit	3714.88
12				

3.1.5 Proceesed file

The second type of sheet has columns like Date, Narration, Amount, and Type. In that type is credit/debit or +/-.

- If the Type column is empty, skip that entry.
- If the credit or debit field has a negative value, add that value to the opposite column (credit to debit and vice versa).

A	B	C	D	E	F	G	H
1		Account Statement					
2	NOW IN SERVICE INDIA PRIVATE LIMITED						
3	SHOP						
4	MAHARASHTRA						
5	.						
6	Surat						
7	GUJARAT						
8	INDIA						
9							
10							
11	Sl. No.	Date	Description	Chq / Ref number	Amount	Dr / Cr	Balance
12		31-01-2023 12:05	MB:RECEIVED FROM PG FBBILL FACEBO		50,260.00	CR	1,00,000.47 CR
13		31-01-2023 00:00	Recd:IMPS/303121		7	30,889.00	CR
14		31-01-2023 00:00	SentIMPS3031177		5	37,565.00	DR
15		31-01-2023 00:00	PG FBBILL FACEBO			-5,600.00	DR
16		31-01-2023 00:00	PG FBBILL FACEBO			-4,800.00	DR
17		31-01-2023 00:00	PG FBBILL FACEBO			-5,300.00	DR
18		31-01-2023 00:00	PG FBBILL FACEBO			5,300.00	DR
19		31-01-2023 00:00	PG FBBILL FACEBO			-4,600.00	DR
20		31-01-2023 00:00	PG FBBILL FACEBO			-5,200.00	DR
21		31-01-2023 00:00	PG FBBILL FACEBO			-4,900.00	DR
22	Opening balance	as on 01/01/2023	INR 18,795.67				
23	Closing balance	as on 31/01/2023	INR 100,000.47				
24	You may call our 24-hour Customer Contact Centre at our number 1860 266 2666						
25	Write to us at Customer Contact Centre, Kotak Mahindra Bank Ltd. Post Box Number 16344, Mumbai 400 013						
26							

3.1.6 User's file

A	B	C	D	E
1	date	description	type	amount
2	31-01-2023	MB:RECEIVED FROM PG FBBILL FACEBO	credit	50260
3	31-01-2023	Recd:IMPS/303121	credit	30889
4	31-01-2023	SentIMPS3031177	debit	37565
5	31-01-2023	PG FBBILL FACEBO	credit	5600
6	31-01-2023	PG FBBILL FACEBO	credit	4800
7	31-01-2023	PG FBBILL FACEBO	credit	5300
8	31-01-2023	PG FBBILL FACEBO	debit	5300
9	31-01-2023	PG FBBILL FACEBO	credit	4600
10	31-01-2023	PG FBBILL FACEBO	credit	5200
11	31-01-2023	PG FBBILL FACEBO	credit	4900
12				
13				

3.1.7 Processed file

Testing:

- First and fourth Column Ignored

Row Number	Reason	Credit/Debit
2,4,8,9,10,12	Negative Sign	Debit
3,5	Cr at end	Credit
6,11	Dr at end	Debit
7	Just Amount	Credit

A	B	C	D	E	F
1	Sl. No.	Date	Description	Chq / Ref	Amount
2	1	05-08-2022	ANITA PANDEY	FCM-220	-16,390.01
3	2	05-08-2022	TEJAS SAGROLI	FCM-220	7252.02 (CR.)
4	3	05-08-2022	SALMA F FREEL	FCM-220	-7,644.00
5	4	05-08-2022	JYOTI SALUJA FI	FCM-220	5684(Cr.)
6	5	05-08-2022	GANESH BABU I	FCM-220	980 (dr)
7	6	05-08-2022	AMRITA BHOUNI	FCM-220	26,460.00
8	7	05-08-2022	PRAHELIKA PAI	FCM-220	-196
9	8	05-08-2022	SREELAKSHMI V	FCM-220	-588
10	9	05-08-2022	SASTRY MALLA	FCM-220	-11,172.00
11	10	05-08-2022	PRIYANKA SINH	FCM-220	7056.01dr.
12	11	05-08-2022	NIDHI JACOB FF	FCM-220	-1,372.00

3.1.8 User's file

A	B	C	D	E
1	date	description	type	amount
2	05-08-2022	ANITA PAND	debit	16390.01
3	05-08-2022	TEJAS SAGRC	credit	7252.02
4	05-08-2022	SALMA F FRE	debit	7644
5	05-08-2022	JYOTI SALUJA	credit	5684
6	05-08-2022	GANESH BAE	debit	980
7	05-08-2022	AMRITA BHC	credit	26460
8	05-08-2022	PRAHELIKA F	debit	196
9	05-08-2022	SREELAKSHM	debit	588
10	05-08-2022	SASTRY MALI	debit	11172
11	05-08-2022	PRIYANKA SI	debit	7056.01
12	05-08-2022	NIDHI JACOE	debit	1372

3.1.9 Proceesed file

Task 2

User Management

In this task I develop a system with three user roles (Admin, User, and Vendors). Admin created as default with a specific UserName & Password.

Key	Value	Description
<input checked="" type="checkbox"/> name	admin	
<input checked="" type="checkbox"/> email	admin@gmail.com	
<input checked="" type="checkbox"/> password	test123	
<input checked="" type="checkbox"/> password2	test123	
<input checked="" type="checkbox"/> role	admin	
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1   "name": "admin",
2   "email": "admin@gmail.com",
3   "password": "$2a$10$HMEscY1UPRstsd.THUwShe4JX08K4spTpZ9yAt5UpP4MfQU0ifXAq",
4   "role": "admin",
5   "_id": "642403ae92e2896d178bdac0",
6   "date": "2023-03-29T09:23:58.867Z",
7   "__v": 0
8
9

```

Status: 200 OK Time: 1513 ms Size:

3.2.1 Create Admin

Users can register to the system. Create middleware to authenticate a user to access API. Send welcome mail when the user registers. Here "NodeMailer" is used for sending mail to the registered users.

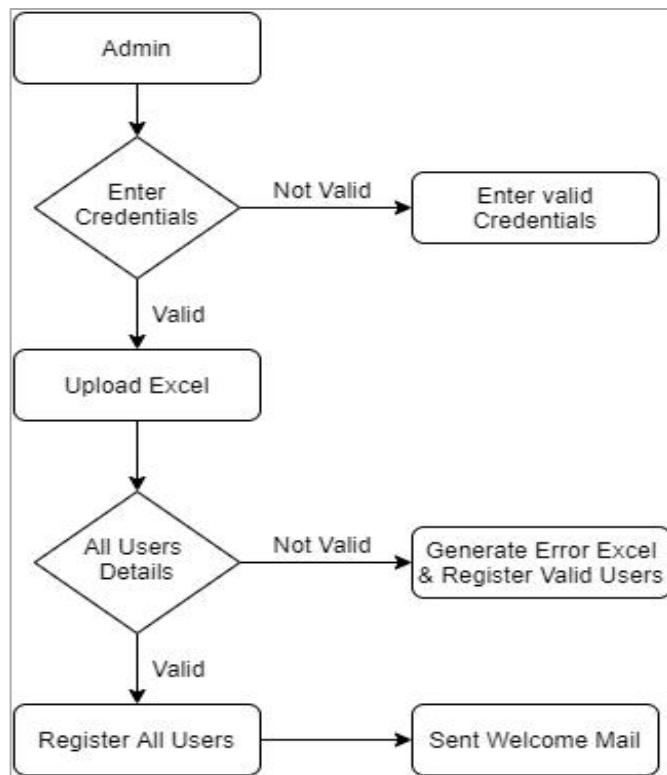


Fig. 3.2.2 Admin Flow-Chart

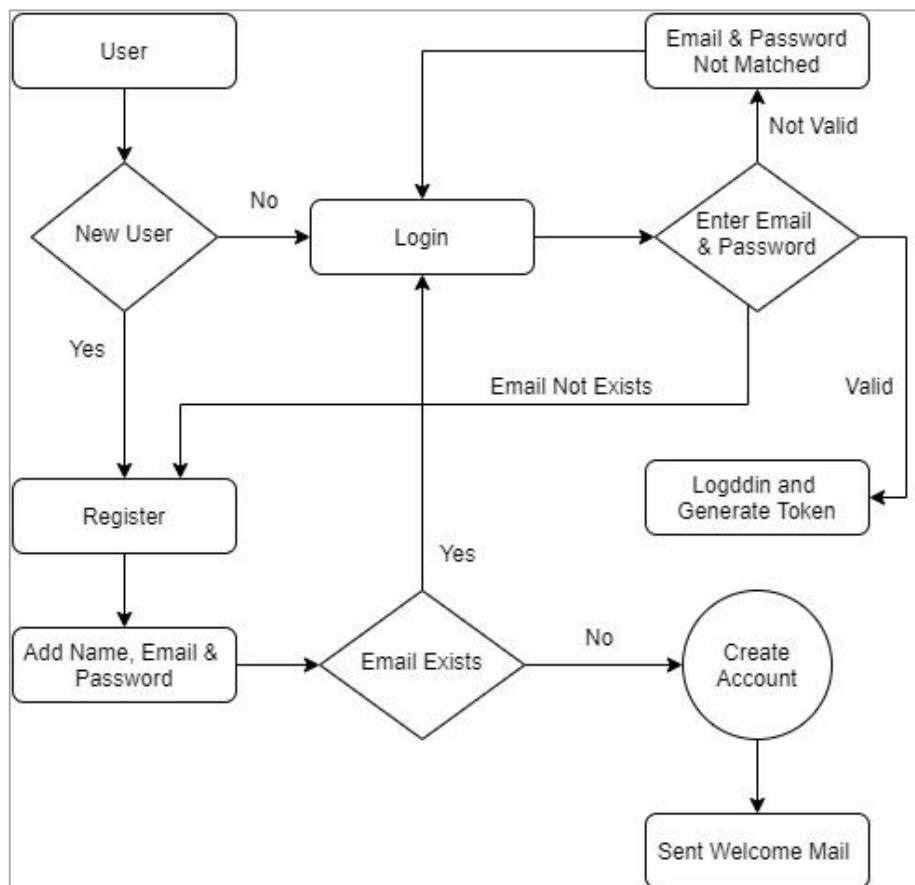


Fig. 3.2.3 User Flow-Chart

The screenshot shows a Postman collection named "APP / register". A POST request is made to "http://localhost:5000/api/users/register". The "Body" tab is selected, showing form-data parameters: name (Princy), email (princygajera28@gmail.com), password (123456), password2 (123456), and role (user). The response status is 200 OK, and the JSON response body is displayed:

```

1
2
3
4
5
6
7
8
9
{
  "name": "Princy",
  "email": "princygajera28@gmail.com",
  "password": "$2a$10$0gvn2u1/tnjdMmpVqhe2T0jZ4S2Xf25pw5.3ju7zeKx3y8/eMwB0y",
  "role": "user",
  "_id": "641acc057ac619beea22410f",
  "date": "2023-03-22T09:36:05.399Z",
  "__v": 0
}

```

3.2.4 User registered successfully

```

in Mongoose 7. Use `mongoose.set('strictQuery', false);` if you want to prepare for
strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
Server up and running on port 5000 !
MongoDB successfully connected
Message sent: <2b2fe2de-13a9-5a1c-c634-dab751abedb3@gmail.com>

```

3.2.5 Mail sent log

The screenshot shows an incoming email in the inbox. The sender is Princy (princygajera121@gmail.com) at 3:06 PM. The recipient is the user (princygajera28@gmail.com). The message content is:

From Princy • princygajera121@gmail.com
 To princygajera28@gmail.com
 Date Mar 22, 2023, 3:06 PM
 Standard encryption (TLS).
[View security details](#)

Hello princygajera28@gmail.com your auto generated password is

3.2.6 Mail sent to new registered user

The screenshot shows a POST request to `http://localhost:5000/api/users/register`. The request body is set to `x-www-form-urlencoded` and contains the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Princy	
<input checked="" type="checkbox"/> email	princygajera28@gmail.com	
<input checked="" type="checkbox"/> password	123456	
<input checked="" type="checkbox"/> password2	123456	
<input checked="" type="checkbox"/> role	user	

The response status is `400 Bad Request` with the message `"email": "Email already exists"`.

3.2.7 Email already exists

Use passport and jwt tokens for authentication of users and vendors. When the user Login to the system, a token is generated for that user session.

The screenshot shows a POST request to `http://localhost:5000/api/users/login`. The request body is set to `x-www-form-urlencoded` and contains the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email	princygajera28@gmail.com	
<input checked="" type="checkbox"/> password	123456	

The response status is `200 OK` and contains the following JSON data:

```

1  "success": true,
2  "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJpZCI6IjY0MWFjYzA1N2FjNjE5YmVlYTYiNDewZlIsIm5hbWUiOijQcmLuY3kiLCJpYXQiOjE2Nzk0NzgwODEsImV4cCI6MTY3OTU3ODA4MX0.
zEc-KZ300rUKPVLDeBA-Vpkv2kxBuS_xYv13tXiuIHo"
3
4
    
```

3.2.8 User Login Successfully

The screenshot shows a POST request to `http://localhost:5000/api/users/login`. The 'Body' tab is selected, showing form-data with 'email' set to `princygajera@gmail.com` and 'password' set to `123456`. The response status is 404 Not Found, with the error message `"emailnotfound": "Email not found"`.

3.2.9 Email not exists

Admin can upload an excel sheet with users and their details. Multer is used for storing uploaded sheets by the admin. API create all the users as per excel sheet data and register all of them into the system.

The screenshot shows a POST request to `http://localhost:5000/api/users/upload`. The 'Body' tab is selected, showing form-data with 'xlsx' set to `test3.xlsx`. The response status is 200 OK, with a JSON response containing two user objects:

```

1 [
2   {
3     "name": "f",
4     "email": "f@gmail.com"
5   },
6   {
7     "name": "g",
8     "email": "g@gmail.com"
9   }
10 ]

```

3.2.10 All users register as per the excel

Generate a password for each excel user and Send welcome mail with an auto-generated password. Save the password into the database with encryption. Here "bcrypt" is used for this purpose.

```

1679478967804-test3.xlsx U X
uploads > 1679478967804-test3.xlsx

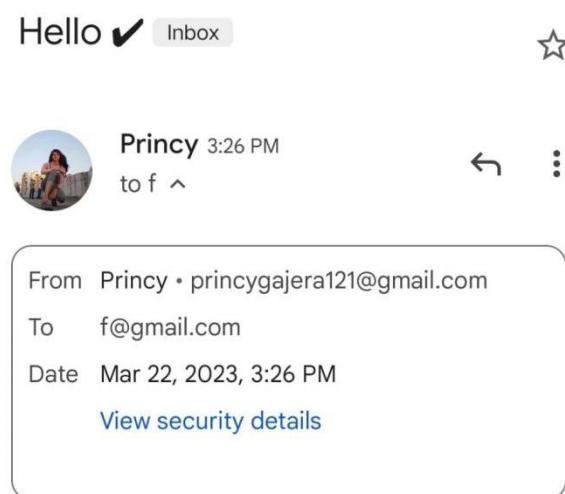
A   B   C
1 name email
2 f     f@gmail.com
3 g     g@gmail.com

Sheet1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

Message sent: <0981f8a3-f3ec-83bb-8f82-7bf7dbc3e62e@gmail.com>
[
  {
    "name": "f",
    "email": "f@gmail.com",
    "password": "$2a$10$bMRYWpfYkJRUVIhpj.JDugKpxW3yPLahzoyF67bLyRB.0E.P6x6",
    "role": "user",
    "_id": new ObjectId("641ad0b77ac619beea22411b"),
    "date": 2023-03-22T09:56:07.978Z
  },
]
  
```

3.2.11 Auto generated password store after encryption



Hello f@gmail.com your auto generated password is &Z4Tsl3@

3.2.12 Mail sent with auto generated password

Validate all the fields of excel. If there is any error or an empty field, generate excel with specific error logs with details of the row and column where the error is generated.

```

1 [
2   {
3     "name": "a",
4     "email": "a@gmail.com"
5   },
6   {
7     "name": "b",
8     "email": "",
9     "ErrorLog": "Email field is required"
10}
11]

```

3.2.13 Error generated in one of the users from excel

In the above sheet, one row has an empty Email field. That's why that user will not register to the system. And an excel sheet is generated with an error log that states, a Email field is required.

A	B
1 name	email
2 a	a@gmail.com
3 b	

A	B	C
1 name	email	ErrorLog
2 a	a@gmail.com	
3 b		("email","Email field is required")

3.2.14 Generated excel with error logs

All sheets are saved to the uploads folder and the sheet's names are appended with the current date. And generated file name has an error keyword appended to the file name.

APP / upload

POST <http://localhost:5000/api/users/upload>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> xlsx	test2.xlsx			
Key	Value	Description		

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 358 ms Size: 389 B Save Response

```

1 [
2   {
3     "email": "c@gmail.com",
4     "name": "",
5     "ErrorLog": "{\"name\":\"Name field is required\"}"
6   },
7   {
8     "name": "d",
9     "email": "d@gmail.com"
10 }
11 ]

```

3.2.15 Error generated in one of the users from excel

In the above sheet, one row has an empty Name field. That's why that user will not register to the system. And an excel sheet is generated with an error log that states, a Name field is required.

A	B
1 name	email
2	c@gmail.com
3 d	d@gmail.com

A	B	C
1 email	name	ErrorLog
2 c@gmail.com		{"name":"Name field is required"}
3 d@gmail.com	d	

3.2.16 Generated excel with error logs

Vendors can add different Categories and Products into that Categories. All categories and products have their unique Id. Category can have different categories into it. That means categories hierarchy is there.

KEY	VALUE	DESCRIPTION	Bulk Edit
categoryName	laptop		
parentId	63e338960f98664adf0242ce		
Key	Value	Description	

```

1
2   "categoryName": "laptop",
3   "parentId": "63e338960f98664adf0242ce",
4   "_id": "641ad2ad7ac619beea22411f",
5   "date": "2023-03-22T10:04:29.006Z",
6   "__v": 0
7

```

3.2.17 Create category

KEY	VALUE	DESCRIPTION
parentId	63e338b10f98664adf0242d0	
productName	vivo	
files	3 files selected	
Key	Value	Description

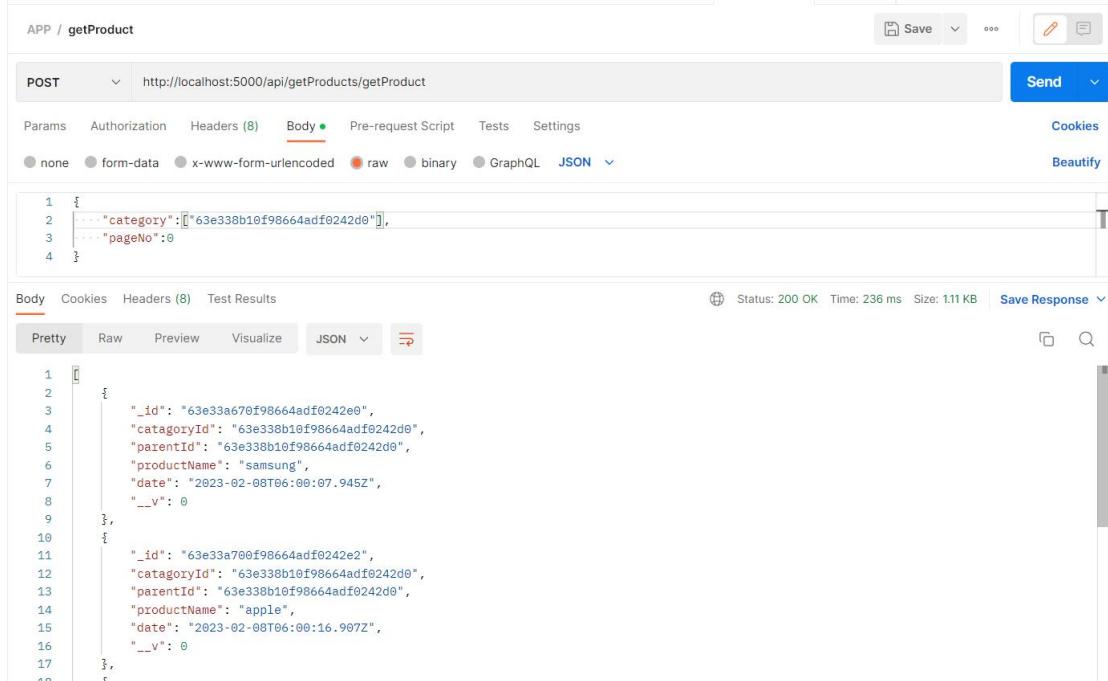
```

1
2   "catagoryId": "63e338b10f98664adf0242d0",
3   "parentId": "63e338b10f98664adf0242d0",
4   "productName": "vivo",
5   "images": [
6     "http://localhost:5000/uploads/1679479719038.jpg",
7     "http://localhost:5000/uploads/1679479719039.jpg",
8     "http://localhost:5000/uploads/1679479719040.jpg"
9   ],
10  "_id": "641ad3a77ac619beea224123",
11  "date": "2023-03-22T10:08:39.044Z",
12  "__v": 0
13

```

3.2.18 Create Product under category

Users can filter out products based on categories and add desired products to their cart.

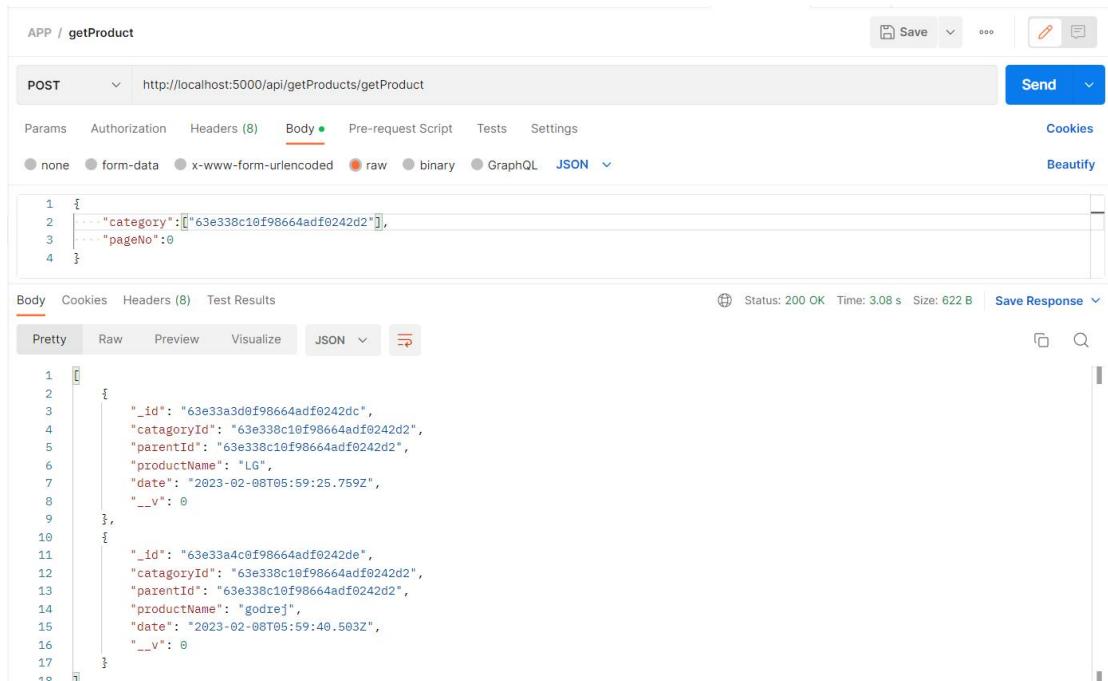


```

POST http://localhost:5000/api/getProducts/getProduct
Body (8)
Params Authorization Headers (8) Body (8) Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   ...
3   "category": ["63e338b10f98664adf0242d0"],
4   ...
5 }
6
7
8
9
10
11
12
13
14
15
16
17
18
```
1 {
2 "_id": "63e33a670f98664adf0242e0",
3 "categoryId": "63e338b10f98664adf0242d0",
4 "parentId": "63e338b10f98664adf0242d0",
5 "productName": "samsung",
6 "date": "2023-02-08T06:00:07.945Z",
7 "__v": 0
8 },
9 {
10 "_id": "63e33a700f98664adf0242e2",
11 "categoryId": "63e338b10f98664adf0242d0",
12 "parentId": "63e338b10f98664adf0242d0",
13 "productName": "apple",
14 "date": "2023-02-08T06:00:16.907Z",
15 "__v": 0
16 },
17
18
```

```

3.2.19 All Products under the Phone category show up



```

POST http://localhost:5000/api/getProducts/getProduct
Body (8)
Params Authorization Headers (8) Body (8) Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   ...
3   "category": ["63e338c10f98664adf0242d2"],
4   ...
5 }
6
7
8
9
10
11
12
13
14
15
16
17
18
```
1 {
2 "_id": "63e33a3d0f98664adf0242dc",
3 "categoryId": "63e338c10f98664adf0242d2",
4 "parentId": "63e338c10f98664adf0242d2",
5 "productName": "LG",
6 "date": "2023-02-08T05:59:25.759Z",
7 "__v": 0
8 },
9 {
10 "_id": "63e33a4c0f98664adf0242de",
11 "categoryId": "63e338c10f98664adf0242d2",
12 "parentId": "63e338c10f98664adf0242d2",
13 "productName": "godrej",
14 "date": "2023-02-08T05:59:40.503Z",
15 "__v": 0
16 },
17
18
```

```

3.2.20 All Products under the Laptop category show up

Task 3

Notifications

In Suvit, we have two types of customers primary and secondary. The Primary customers purchase our plans, while their respective customers are called secondary customers.

When Primary customers want to send some message to their secondary customers, they can directly connect with them by using push notifications and the email service of the suvit app.

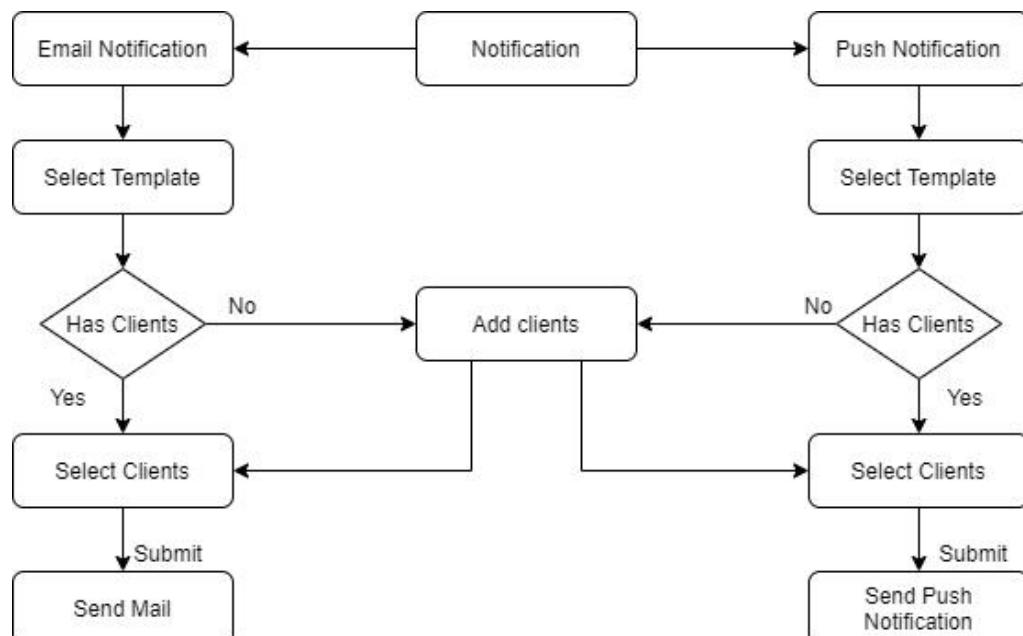


Fig. 3.3.1 Notifications Flowchart

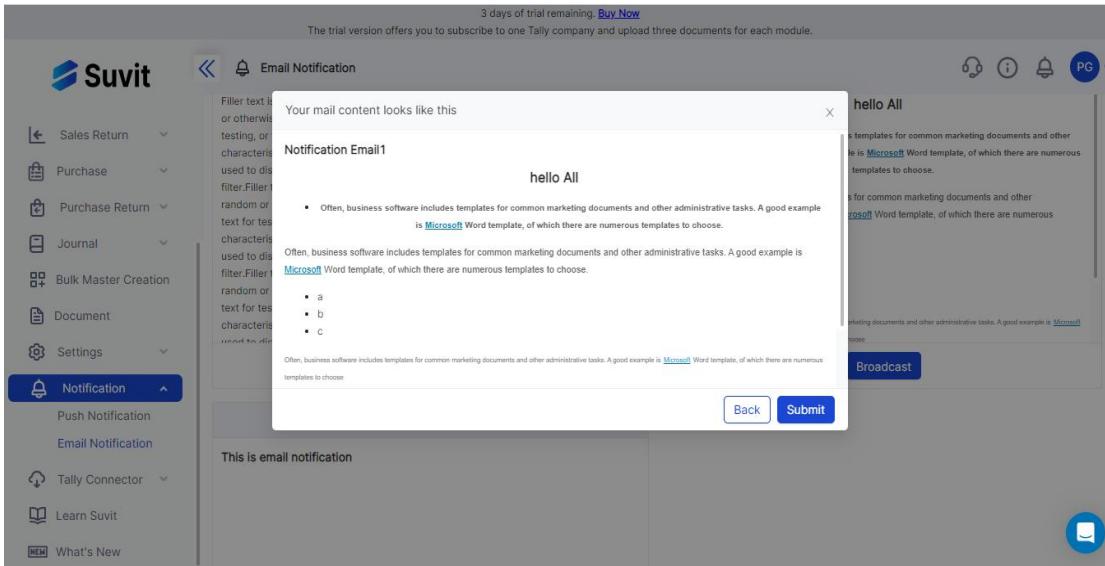
3.1: Mail Notifications

For this purpose, we implement a mail service for the primary customer to send direct mail to the secondary customers with “Nodemailer”.

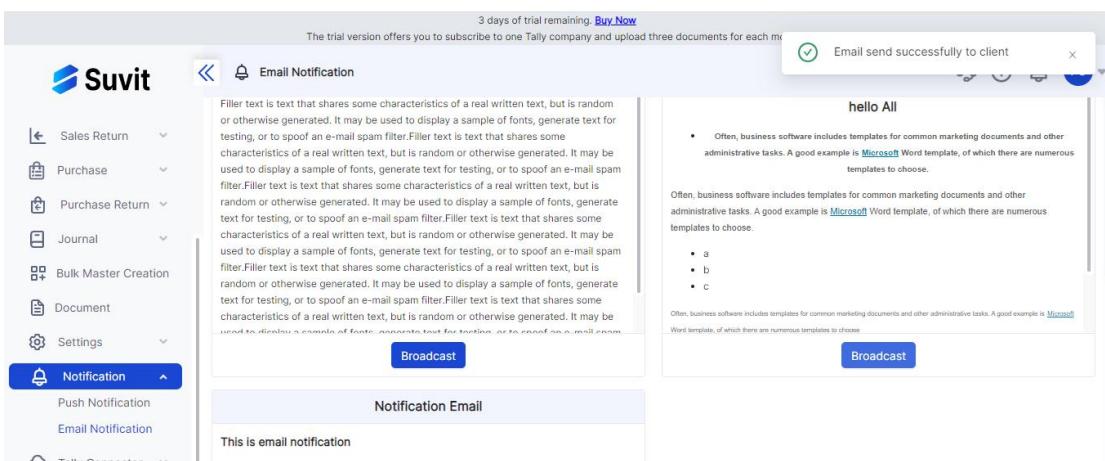
3.3.2 Different templates to send mail

There are templates available to send mail, and from that client can choose one template and respective secondary users and sends mail with just one click.

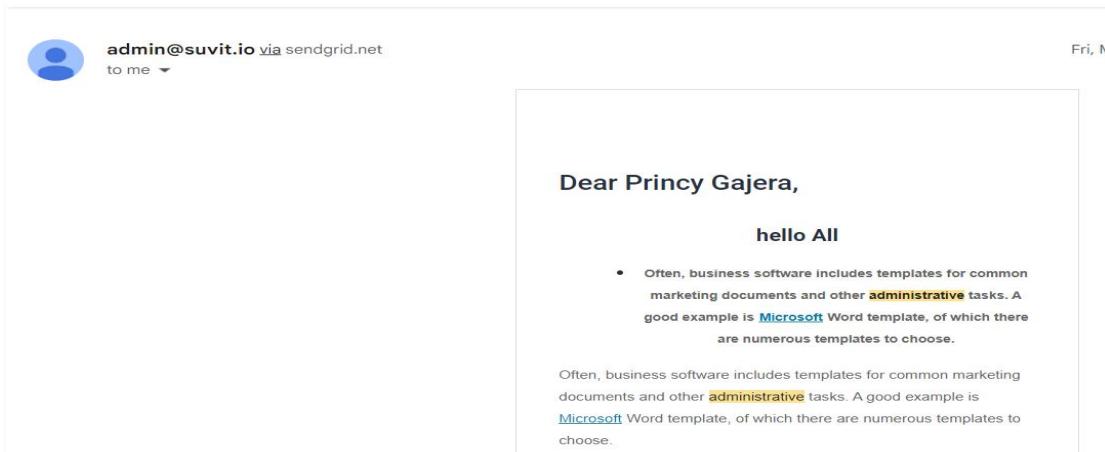
3.3.3 Select all clients to whom the user wants to send mail



3.3.4 Mail Preview



3.3.5 Mail sent confirmation



3.3.6 Mail receive by client

3.2: Push Notifications

Like mail service, if the client wants to send mobile notifications to their customers, Suvit has a Push-Notification service.

Here “OneSignal” Push notification service is used to fulfill our requirements. The client can send mobile notifications to users who have Suvit mobile app. Same as the mail service, the client has templates for Push-Notification and sending a message with just a click.

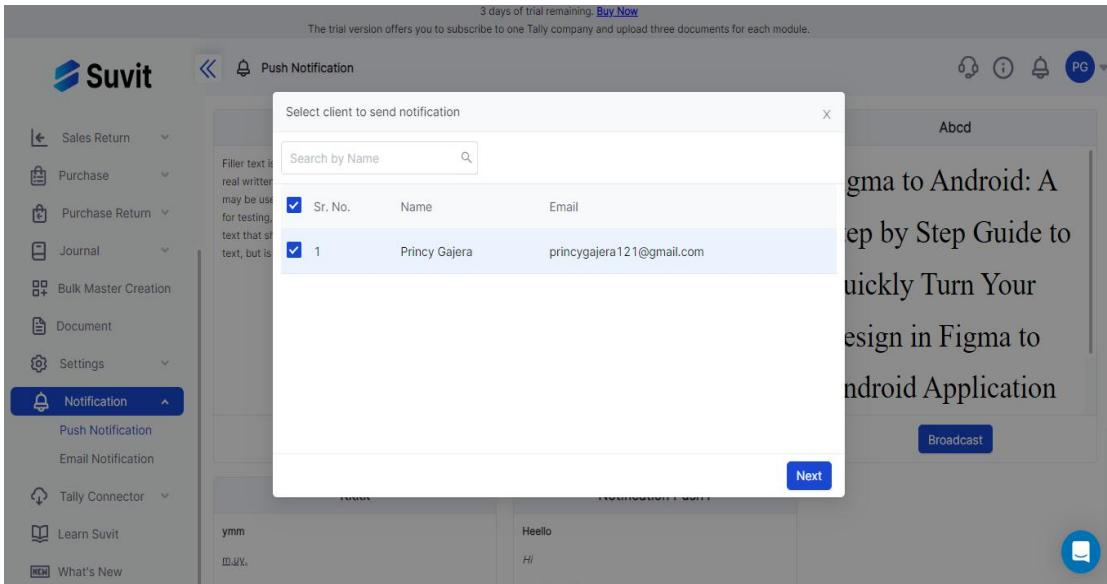
The screenshot shows the Suvit application interface with the 'Push Notification' module selected. On the left sidebar, under 'Notification', 'Push Notification' is highlighted. The main area displays several broadcast options:

- Spam Filter:** A text input field containing placeholder text about a spam filter, with a 'Broadcast' button below it.
- Filter:** A text input field containing placeholder text about a filter, with a 'Broadcast' button below it.
- Abcd:** A text input field containing 'Abcd', with a 'Broadcast' button below it.
- Figma to Android: A Step by Step Guide to Quickly Turn Your Design in Figma to Android Application With DhiWise:** A text input field containing a detailed description of the guide, with a 'Broadcast' button below it.

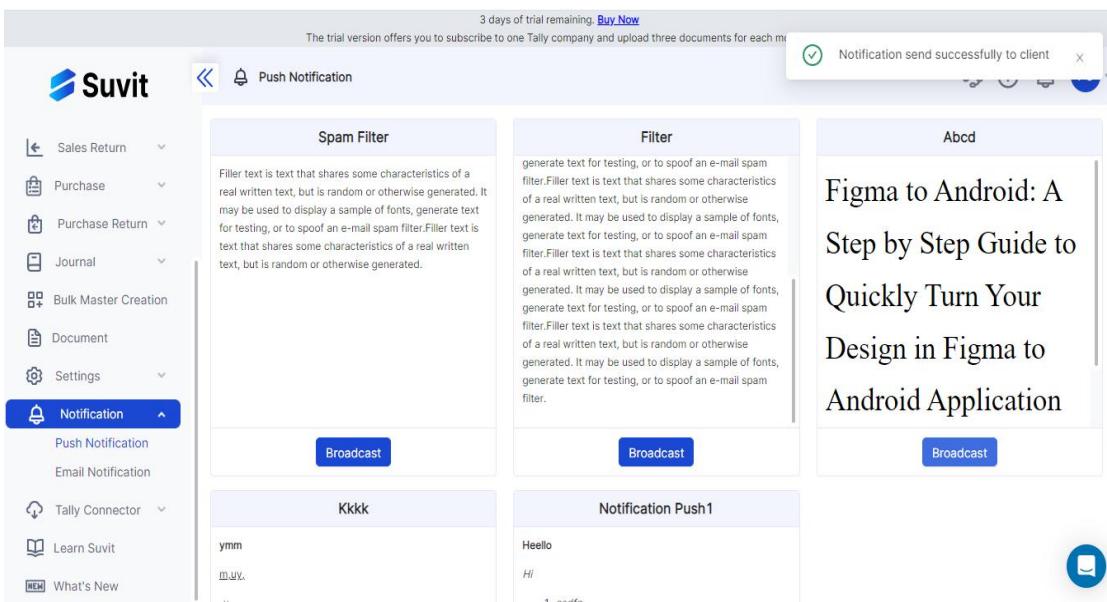
3.3.7 Different templates to send notification

The screenshot shows the Suvit application interface with the 'Push Notification' module selected. A modal window titled 'Your notification content looks like this' is open, displaying the text from the 'Figma to Android' template. The modal includes a 'Back' button and a 'Submit' button. In the background, the main interface shows the same broadcast options as the previous screenshot.

3.3.8 Select notification template



3.3.9 Select all clients to whom the user wants to send notification



3.3.10 Notification sent confirmation

Task 4

Hub Spot

HubSpot is a tool that seamlessly connects your data, teams, and customers on one CRM platform that grows with a business. We integrate Suvit with HubSpot.

We add all the contacts and companies to HubSpot. Whenever users register to the website, we add their details to the HubSpot in our contact list. If any user updates the user profile, those changes also reflect on HubSpot.

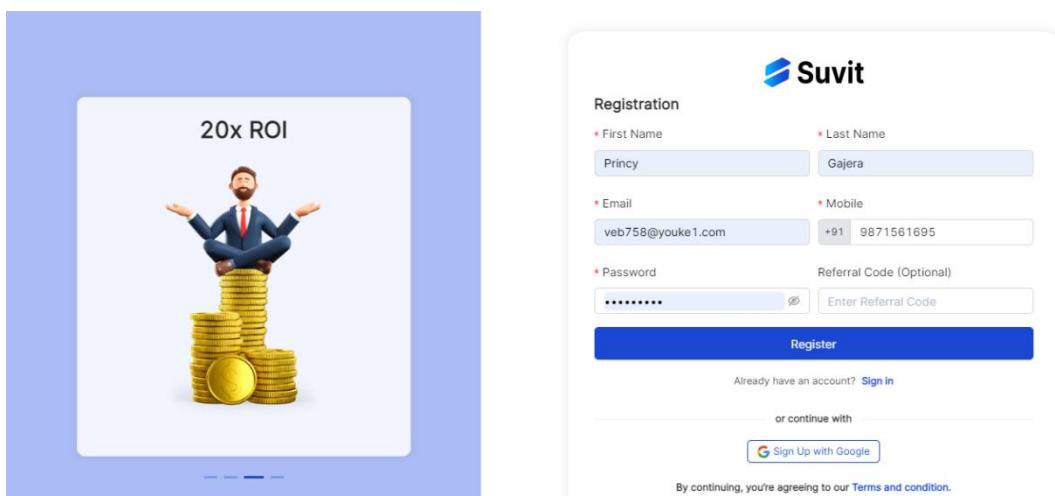


Fig. 3.4.1 User Register to the Suvit

Testing:

NAME	EMAIL	PHONE NUMBER	CONTACT OWNER	PRIMARY COMPANY
Princy Gajera	vcb758@youke1.com	+91 9871561695	Unassigned	Youke1

Fig. 3.4.2 User details added to the HubSpot Contacts list

When users purchase any plan from Suvit and become Live customers from trial or expired one, add their company details to the HubSpot company list.

According to the company's domain name, all the contacts under them also show up in HubSpot. Also, we can update or delete the company from HubSpot.

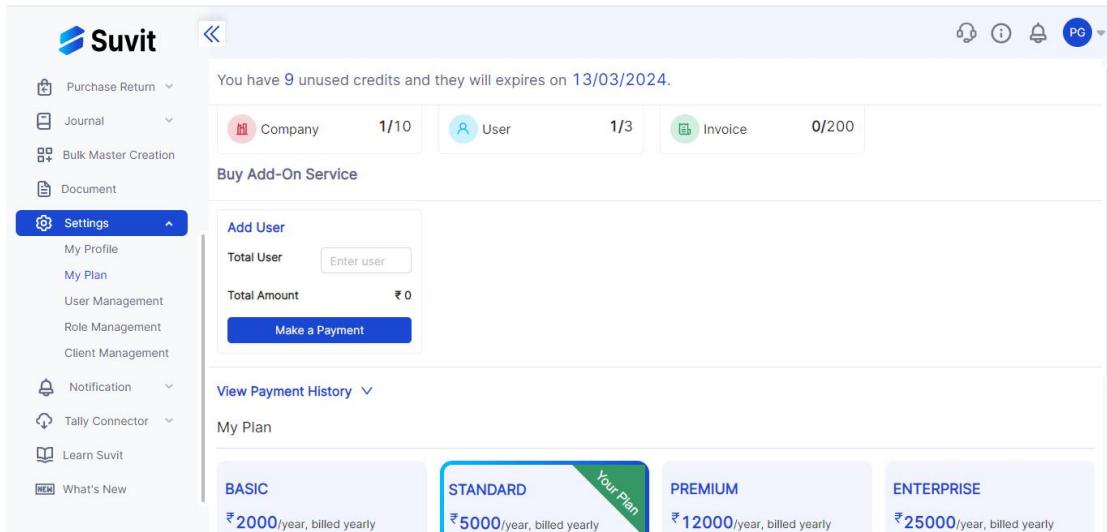


Fig. 3.4.3 User purchase plan and become Active Client

Testing:

As we can see When a user purchases a plan and become a live user, their details are added to the HubSpot company list and their domain name will become a Company name.

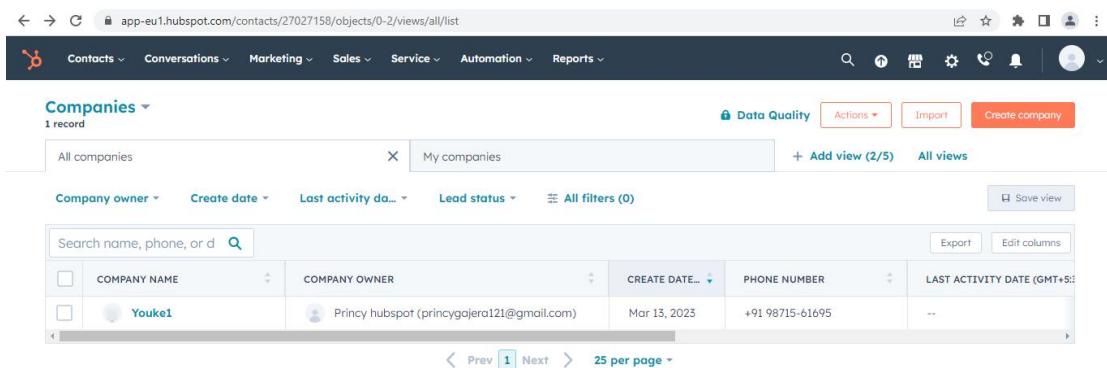


Fig. 3.4.4 Clients details added to the HubSpot Company list

Task 5

Split Company

If a user has too many records in one company and wants to split them into two different companies then this feature is useful.

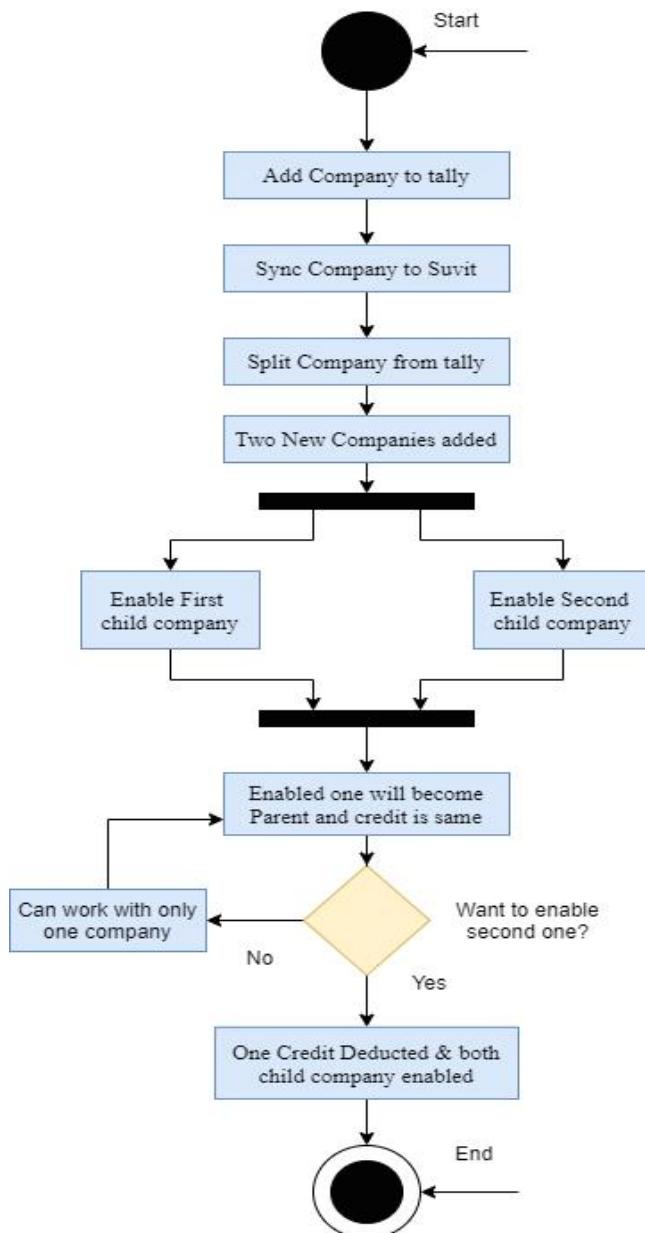


Fig. 3.5.1 Activity Diagram for split company

In this feature, a user can split one company into two from the Tally. After splitting two companies were added with the same name as the parent company.

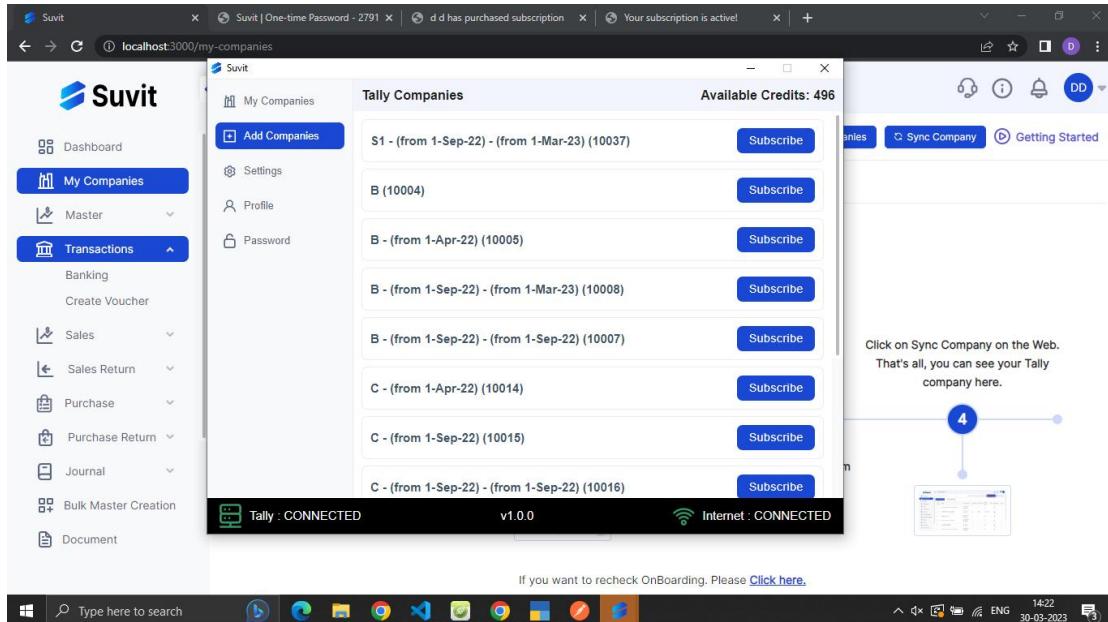


Fig. 3.5.2 List of all Companies in Electron

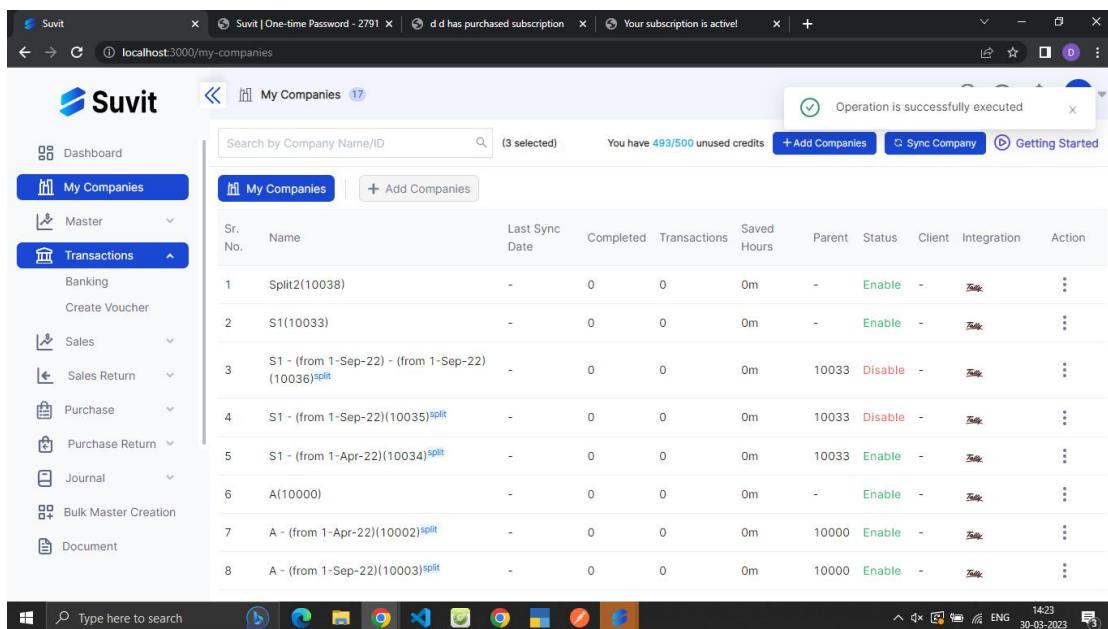


Fig. 3.5.3 Subscribe company to Suvit

We have one financial year from 1st April to 31st March. So the first child company is created from 1st April to 30th September. And the second one is from 1st October to 31st March.

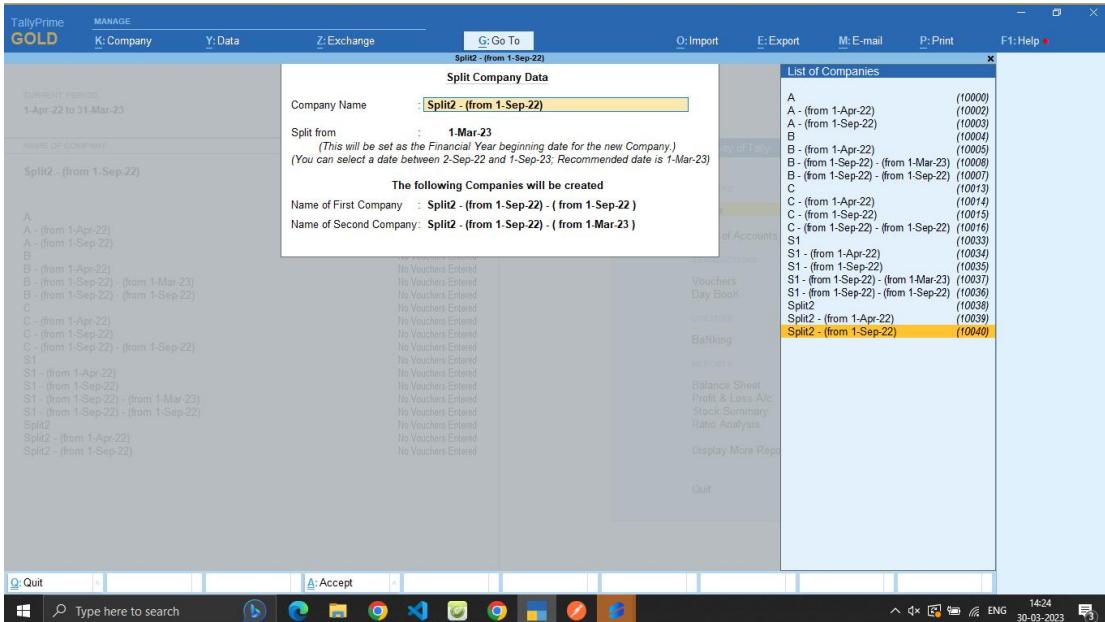


Fig. 3.5.4 Split company from Tally

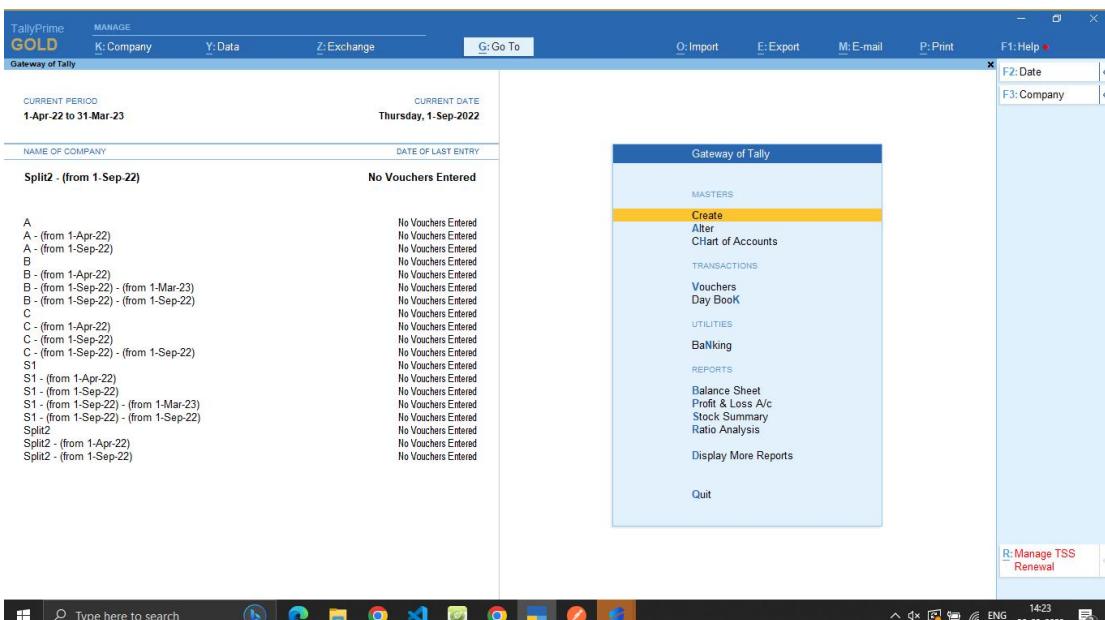


Fig. 3.5.5 Two new companies added with the same name &different dates

In the Suvit both newly created companies show up, and their status is disabled at first. We need to subscribe to them first, after subscribing parent company is disabled and new companies are enabled. Whichever child company is enabled first will become a parent.

Sr. No.	Name	Last Sync Date	Completed	Transactions	Saved Hours	Parent	Status	Client	Integration	Action
1	Split2(10038)	-	0	0	0m	-	Disable	-		
2	Split2 ~ (from 1-Sep-22)(10040) split	-	0	0	0m	10038	Disable	-		
3	Split2 ~ (from 1-Apr-22)(10039) split	-	0	0	0m	10038	Disable	-		
4	S1(10033)	-	0	0	0m	-	Enable	-		
5	S1 ~ (from 1-Sep-22) - (from 1-Sep-22)(10036) split	-	0	0	0m	10033	Disable	-		
6	S1 ~ (from 1-Sep-22)(10035) split	-	0	0	0m	10033	Disable	-		
7	S1 ~ (from 1-Apr-22)(10034) split	-	0	0	0m	10033	Enable	-		
8	A(10000)	-	0	0	0m	-	Enable	-		

Fig. 3.5.6 All three companies shows in Suvit

We have one credit for each company. This means when we create the first parent company one credit was deducted. Now when we enable first child company our credit will remain the same. But at the time of the second company, one credit is deducted.

Testing:

Sr. No.	Name	Last Sync Date	Completed	Transactions	Saved Hours	Parent	Status	Client	Integration	Action
1	Split2(10038)	-	0	0	0m	-	Enable	-		
2	Split2 ~ (from 1-Sep-22)(10040) split	-	0	0	0m	10038	Disable	-		
3	Split2 ~ (from 1-Apr-22)(10039) split	-	0	0	0m	10038	Disable	-		
4	S1(10033)	-	0	0	0m	-	Enable	-		
5	S1 ~ (from 1-Sep-22) - (from 1-Sep-22)(10036) split	-	0	0	0m	10033	Disable	-		
6	S1 ~ (from 1-Sep-22)(10035) split	-	0	0	0m	10033	Disable	-		
7	S1 ~ (from 1-Apr-22)(10034) split	-	0	0	0m	10033	Enable	-		
8	A(10000)	-	0	0	0m	-	Enable	-		

Fig. 3.5.7 Parent company enabled

As we can see When a user enables the first company, credit is the same as before but after enabling the second child company, one credit is deducted.

Sr. No.	Name	Last Sync Date	Completed	Transactions	Saved Hours	Parent	Status	Client	Integration	Action
1	Split2(10038)	-	0	0	0m	-	Enable	-	Test	⋮
2	Split2 - (from 1-Apr-22)(10039)split	-	0	0	0m	10038	Disable	-	Test	⋮
3	Split2 - (from 1-Sep-22)(10040)split	-	0	0	0m	10038	Enable	-	Test	⋮
4	S1(10033)	-	0	0	0m	-	Enable	-	Test	⋮
5	S1 - (from 1-Sep-22) - (from 1-Sep-22)(10036)split	-	0	0	0m	10033	Disable	-	Test	⋮
6	S1 - (from 1-Sep-22)(10035)split	-	0	0	0m	10033	Disable	-	Test	⋮
7	S1 - (from 1-Apr-22)(10034)split	-	0	0	0m	10033	Enable	-	Test	⋮
8	A(10000)	-	0	0	0m	-	Enable	-	Test	⋮

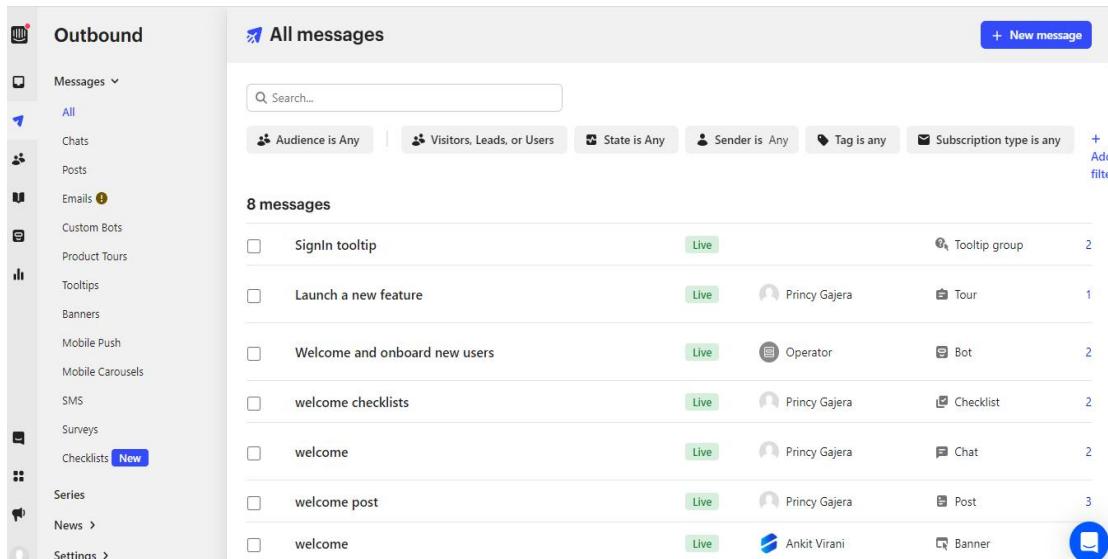
Fig. 3.5.8 One child company enabled

Task 6

Intercom

Intercom is an AI automation tool, that provides messenger, chats, and many more services.

We integrate it with the suvit app. First, we integrate messenger into Suvit. Then Chats, Posts, Tool-tips, Chat-bots and product tours, are integrated.

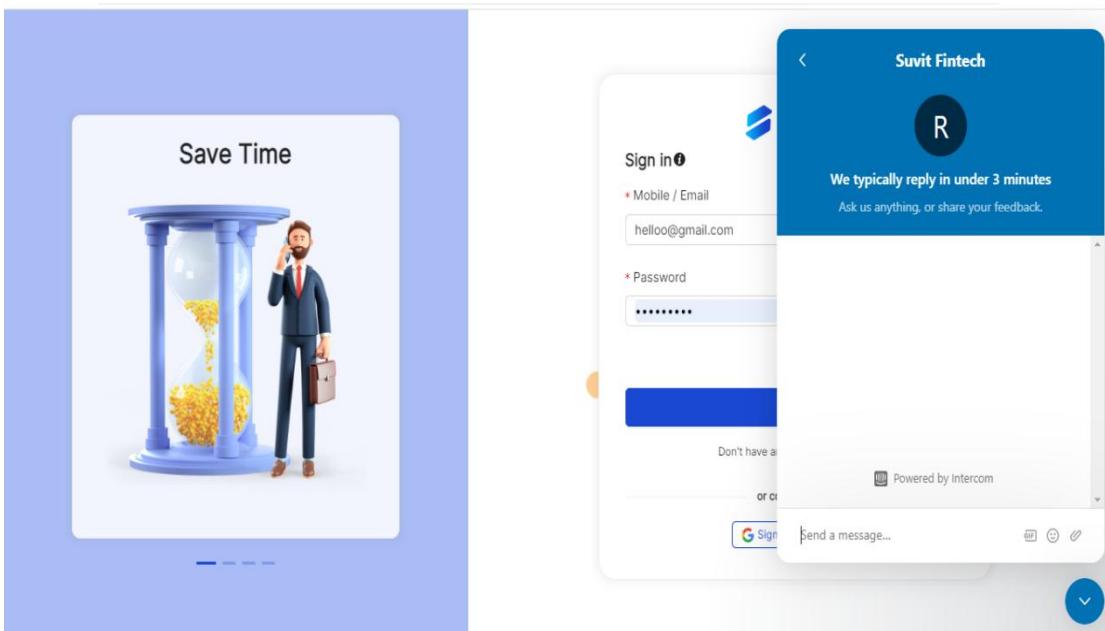


The screenshot shows the Intercom Outbound interface. On the left, there's a sidebar with various categories: Messages (All, Chats, Posts), Emails (Custom Bots, Product Tours, Tooltips, Banners, Mobile Push, Mobile Carousels), SMS, Surveys, Checklists (New), Series, News >, and Settings >. The main area is titled "All messages" and shows a search bar and several filter options: Audience is Any, Visitors, Leads, or Users, State is Any, Sender is Any, Tag is any, Subscription type is any, and an "Add filter" button. Below this, it says "8 messages" and lists the following items:

Message	Status	Sender	Type	Count
SignIn tooltip	Live		Tooltip group	2
Launch a new feature	Live	Princy Gajera	Tour	1
Welcome and onboard new users	Live	Operator	Bot	2
welcome checklists	Live	Princy Gajera	Checklist	2
welcome	Live	Princy Gajera	Chat	2
welcome post	Live	Princy Gajera	Post	3
welcome	Live	Ankit Virani	Banner	

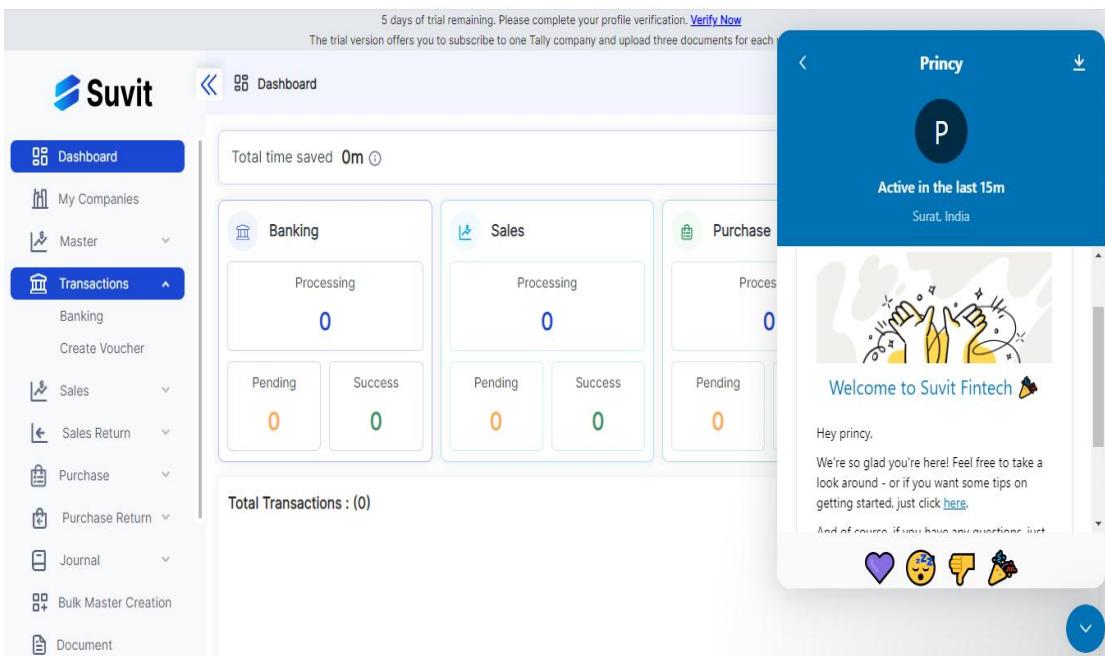
3.6.1 Intercom implemented feature

Intercom powers customer communications, delivering conversions, customer engagement, and personalized support. The Messenger is the heart of this strategy.

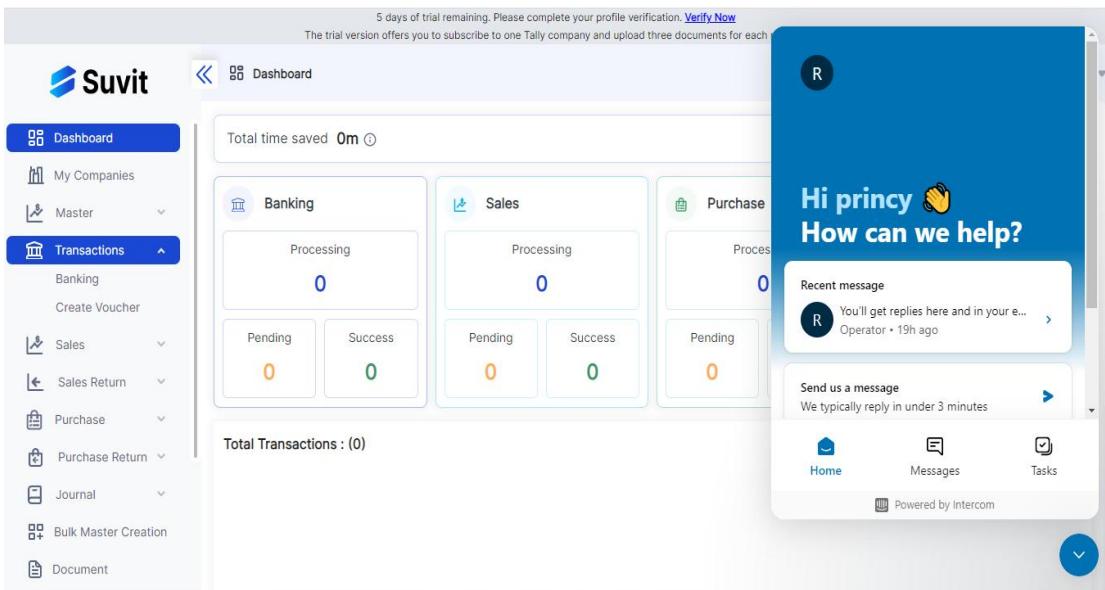


3.6.2 Messenger without Registration

We have almost all of our support channels integrated into one system, making it easy for us to manage our support, as well as take a data-driven approach to continuously improving our service.

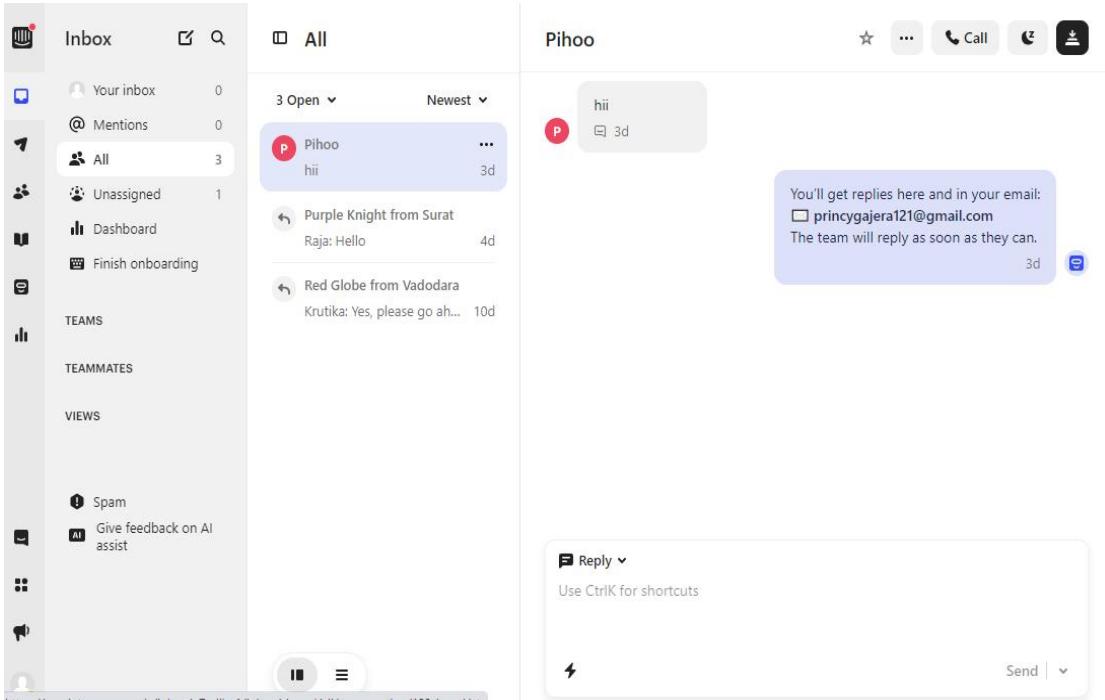


3.6.3 Welcome Post to Registered User

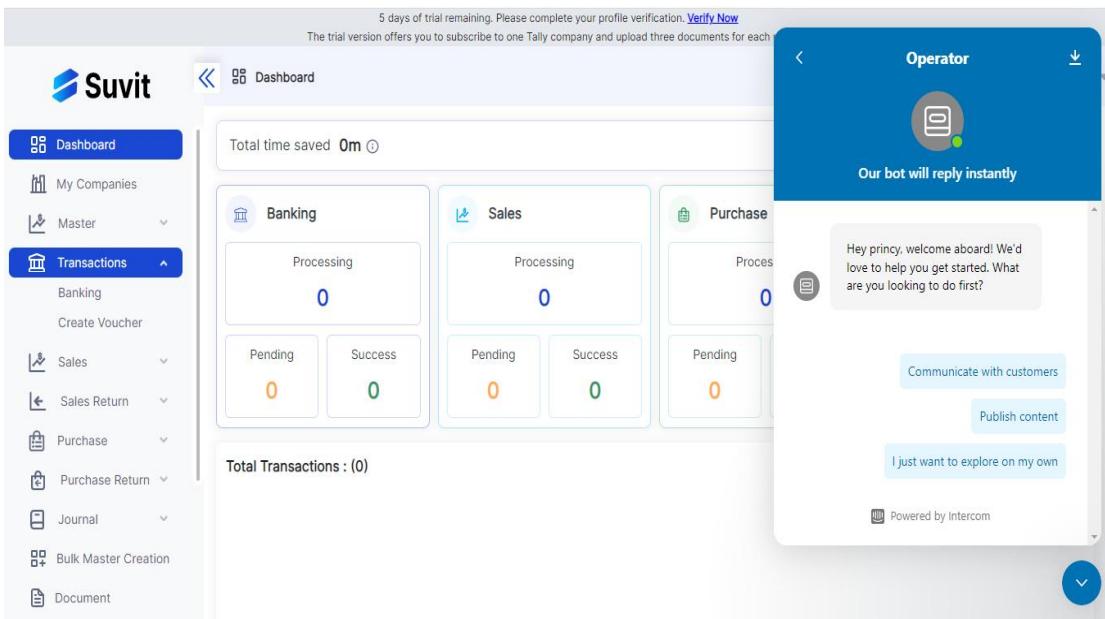


3.6.4 Chats Integration

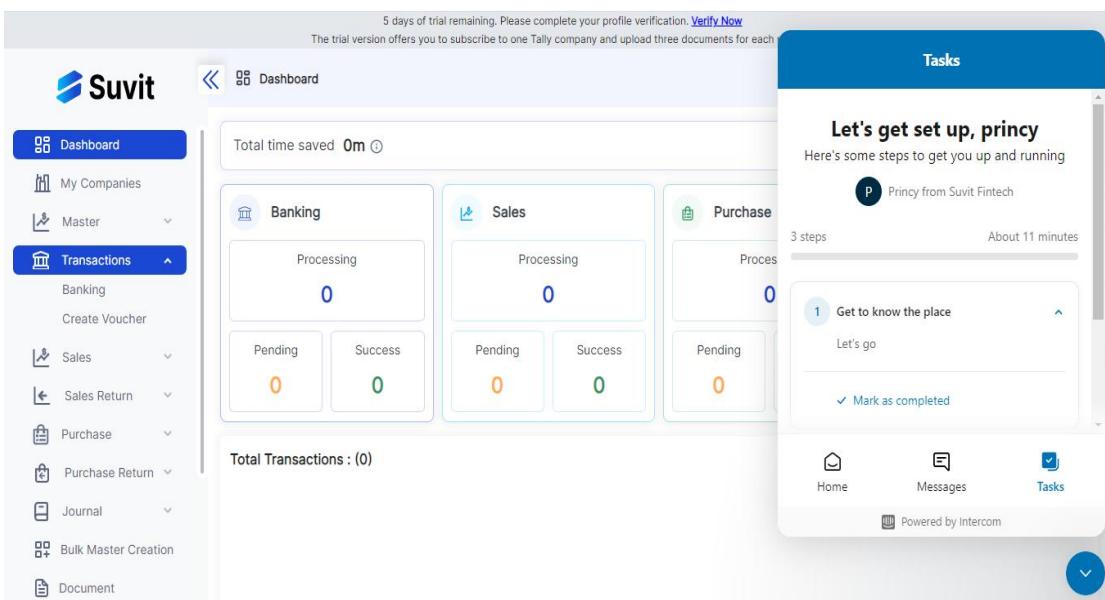
With Live Chats, user can directly chat in the product. We make it easy for customers to get help right when—and where—they need it.



3.6.5 Message receive in intercom inbox



3.6.6 Custom Bots Integration



3.6.7 Checklists Integration

CONCLUSION

In summary, Suvit is a platform designed to integrate various accounting and tax programs into a single office environment. With its advanced technologies such as Artificial Intelligence and Machine Learning, Suvit aims to shorten the data lifecycle for a better response and bridge the gap between taxation, accounting, banking, auditing, professionals, and users involved in such sectors. The platform offers an accounting automation solution that reduces clerical and manual tasks and enables better communication, collaboration, and opportunities to work on futuristic approaches.

The completion of all the tasks marks a significant achievement in my programming journey. With the help of advanced technologies such as NodeJs and ExcelJs, I was able to create highly functional modules for Suvit. In Document Processing, we implement an algorithm in NodeJs instead of Python, which is quite a difficult and research-based Process. But it is a useful feature for accounting software. In Notifications, we have a plan to introduce SMS and WhatsApp services for users. While in HubSpot, we will add Primary Associations to the company.

BIBLIOGRAPHY

<https://www.intercom.com/>

<https://www.hubspot.com/>

<https://developers.hubspot.com/docs/api/overview>

<https://www.mongodb.com/atlas/database>

<https://www.mailinator.com/>

<https://onesignal.com/>

<https://nodemailer.com/about/>

<https://www.postman.com/>

<https://jwt.io/>

<https://www.passportjs.org/>

<https://www.npmjs.com/package/bcryptjs>