

project introduction:

This SAAS type web application acts as user interface for AI agent built in n8n .

For both ,

1. Client : the business which purchase ai chatbot services from us to integrate ai chatbots in their system or in their social media channels ,
2. Admin : our company , which provide those ai chatbot services (building , maintaining , deployment , ...) to business (clients) ,

So that admin(we) do not have to create and manage separate interface and database for each client , which becomes very complex to manage and costly also ,

Using this SAAS based web application we can provide interface to client by just creating new account , so that we can focus on quality of the core agent in n8n ,

And now admin can provide interface to client and can get database configured so that he can just focus on core agent

So main propose of this project is to provide clients interface to ,

1. See message history (incoming and outgoing messages from bot)
2. See analytics of the chatbots
3. To manage knowledgebase (adding or deleting documents from KB so that KB will be updated in real time)
4. Manage payments (paying initial building fee(developmetn fee) and monthly service fee (subscription fee) through application so that admin do not have to manage all the payments manually)
5. See customer contact information (contact information of the users who have messaged till now)
6. Get some help from admin (so that admin can mange all the customer service in that application which reduce the complexity of communicating to clients in different messaging platforms)
7. Get some minor services

And provide admin interface to :

1. Monitoring all the clients at same place
2. Getting payments managed by the system
3. Getting analytics of all the clients and chatbots
4. Manage each client and chatbot
5. Managing different subscription plans
6. Other settings

Project overview :

This app have two frontends :

1. Client UI
2. Admin UI

Both will use same backend , backend will use role based routing to manage permission to different services and resources

But the database used by this backend will also be used by n8n agent ,

Like two frontends sharing their resources ,

I mean some resources are created by admin and used by client and vice versa

Similarly some resources are created by this backend in the database will used by n8n agent and resources created by n8n agent will be used by this backend

For example :

For KB , this backend will upload , chunk, embedded and will store in vector table ,

Which vector will be used as KB by the n8n agent ,

And the messages will be stored in message history table by n8n agent as it responds incoming messages and this backend will use those messages from table and show on client UI

Whole combination will contain

Two frontends

Backend

Database

N8n agent

But the n8n agent will be built as per client requirement later , since we can connect n8n agent to our interface app by just using this same database in n8n agent ,

So in this application we are building

Both Frontends

Backend and configuring database ,

scope of this application:

I already mentioned that scope of this application is to build interface for client and admin , and to create resources(database) , which also will be used in the n8n agent also to connect main agent to this interface application

N8n agent is out of scope of this application,

Here creating resources (database) is not separate task ,
it is the database required for this interface backend we will just use it in n8n also

So this application has two frontends and one backend ,
And the scope of this application is the scope of this backend:

So **Scope of this backend** :

1. message history, (n8n agent will write all the message to database table and this application will just render them to its frontends)
2. analytics of the chatbot , total message , message through any channel , ...many more
3. Manage payment , (paying monthly or yearly subscription or cost occurred per month or)
4. customer contact information , (the contact details of the customer who have messaged) (added to database by n8n agent , this application will just render in frontends)
5. Manage knowledgebase : adding , editing or deleting the documents which will be used as context (this application will handle adding document in vector tables (all the process related to it , chunking , embedding , .. to add new document to KB) , deleting any document so that all the vectors of that document will be removed from the vector table , so that client can see what context is provided and manage the context but retrieving context from that vector table is not its scope , all the retrieval part will be done by n8n agent itself , since we will use the same database from this backend and also from n8n backend ,
6. Different operations by admin or clients

tech stack :

1. Frontend : React with typescript using vite and tailwind CSS for design
2. Backend : Express with typescript
3. Database : postgres (neon)
4. Vector db : PG vector extension
5. Embedding model : openai embedding model
6. Payment gateway : Khalti and esewa

detailed technical overview :

We have two frontends

1. Admin UI
2. Client UI

Both frontends use same backend ,

All three, will run in different ports,

backend will use role based routing to manage permission to different services and resources

Postgres neon database will used for both data storage and vector store

PG vector extension will be used for vector storage ,

Payment gateway :

Since most of our clients are from Nepal , we will use two payment gateways popular in nepal

1. Khalti : use web checkout of Khalti
2. Esewa : use Epay

For both use test credentials , from their official documentation

Khalti : <https://docs.khalti.com/khalti-epayment/>

Esewa : <https://developer.esewa.com.np/>

Real credentials will be added later in production

Authentication :

During registration following fields are necessary to be filled for client ,

1. Full Name
2. User name
3. Email
4. Password
5. Mobile number
6. Company name

Verify the email address

Add forget password feature ,

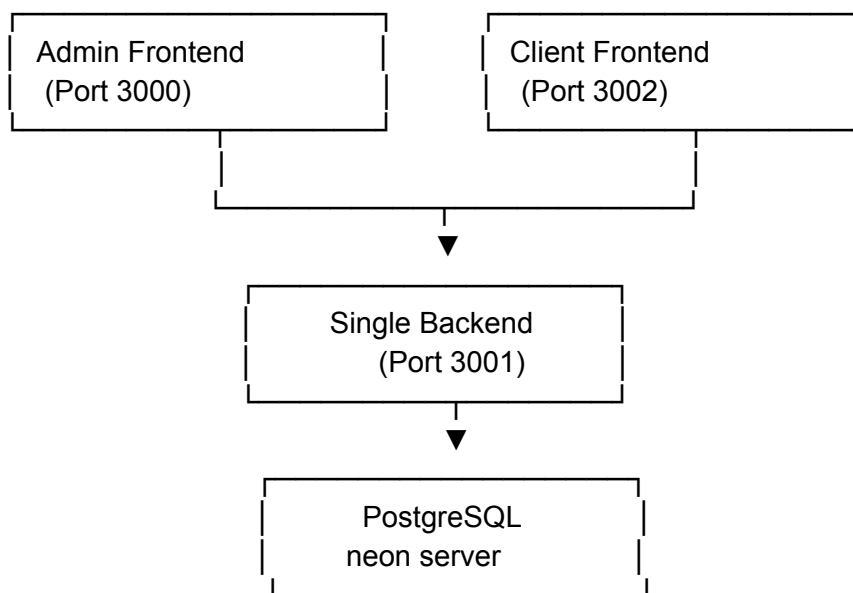
Use best fitting built in modules like [passport.js](#) so that latest authentication features can be added without any complexity

For both client and admin , sign in will be through Username , email and password ,

Login session will be of 1 month

There is no sign up for admin because there will only one admin account and it will added during deployment ,but admin can change user if , email , password , name ...for security

Project structure will like following :



subscription plans :

Subscription are has two types of cost ,

1. Development plans (One time development cost)
2. subscription plans (Long term retaining service cost) ,

development plans :

Package 1 – Website AI Chatbot

Price: ~ NPR 30,000

Features:

- AI chatbot integrated into your website
- Admin dashboard to manage and update the knowledge base
- Access to conversation history for review and analysis

Best For: Businesses seeking a reliable AI assistant to handle customer queries on their website.

Package 2 – Multi-Platform AI Chatbot

Price: ~NPR 55,000

Features:

- AI chatbot integrated into:
 - Website
 - Facebook Messenger
 - Instagram Direct Messages
 - WhatsApp
 - Admin dashboard for knowledge base updates and conversation history management
- Best For:** Businesses wanting consistent customer support across multiple communication channels.

Package 3 – Multi-Platform AI Chatbot with HITL (Human-in-the-Loop)

Price: ~ NPR 70,000

Features:

- All features from Package 2
- Human-in-the-Loop capability:

- Allows a human agent to take over conversations when needed
- Seamless transition between AI and human agent
Best For: Businesses that need the speed of AI with the option for human intervention in complex or sensitive cases.

Package 4 – Multi-Platform AI Chatbot with HITL, Meeting Scheduling, and CRM Integration(google sheets)

Price: ~ NPR 90,000

Features:

- All features from Package 3
- Meeting scheduling directly through the chatbot
- Lead capture into CRM for automated follow-up and tracking
Best For: Businesses looking to combine customer support, appointment booking, and lead generation in one system.

Package 5 – Custom AI Solutions

Price: Based on requirements

Features:

- Custom AI chatbot or agent development based on your unique business needs
- Integration with any required platform or internal system
- Industry-specific workflow automation
Best For: Businesses with specialized requirements beyond standard chatbot features.

subscription plans :

Retaining cost :

We offer mainly three types of subscription systems :

1. **Per message based** : 1.5 npr per message

(pay only as many messages are generated by AI
Will be paid half-yearly)

2. Monthly packages :

Npr 5k: 5k messages “ “
Npr 8k: 9k messages “ “
Npr 10k : 12k messages “ “
Npr 15k : 18k messages “ “
For 18k+ messages (negotiable)

- will be paid monthly in advance
- If messages exceed the subscription plan then per message cost will 1.5 npr per additional message
- If any company bought the package 5k per month but only 4k messages consumed then remaining 1k can not be used in next month (i.e. credit should be consumed the period)
- Same packages can be made for yearly doing 12* to the monthly package (for yearly package will get 10% discount to respective monthly rate)

3. **Recharge system** : rate 1.25 npr per message

(will be like we recharge to call in our phone , paid in advance and can use as long it covers the cost of messages)

During registration client will register through registration pipeline without selecting any package or subscription plans ,
Like register for free ,

Now user can login to client dashboard ,

Now admin can see the new registered user and will contact the client for further setup ,
And will guide the client for choosing development plans ,
For custom plan ,
As per agreement , admin will set chatbot development cost ,

Whichever the development plan
30% of which will be paid initially and remaining at the end of the project,
For all payments application will have payment gateway
After payment portal itself will generate invoice,

Now admin will start building the chatbot

For All communication between client and admin there will be messaging portal so that admin and client can communicate with each other , and client can provide required credentials to admin
And after finishing the setup admin will upgrade it as finished project and from 7 days of that day client have to pay remaining amount otherwise chatbot will temporarily be suspended

After paying the remaining amount, client will choose subscription plan ,
And subscription will start from that day ,

Client will given chance to change their subscription plan after every six month ,

Frontend - client UI :

This is the interface from where our clients will access their chatbot KB and customers data and messages , also will some sort of basic setting for bot ,

Pages in client UI :

1. **Dashboard**
2. **Messages** : there will be separate pages for each channel .
i mean if chatbot is integrated to whatsapp , website and fb messenger then there will be 3 pages , for each channel .
Use similar theme color for respective channel ,
I mean for whatsapp messages page use dark green color

Design of the pages will also be similar to respective application , so that client will be familiar very fast with the interface
3. **Analytics** : different types of analytics related the chatbot
4. **KB- Documents** (client can add or delete document from KB , on deleting document all the chunks or vectors of that document will be deleted from the vector store , so give document id to all the chunks)
5. **Users (customers)or CRM** : contact information of all the customers who has messaged till now .
Name ,
social media id (which social media channel they used to message) and
phone number (it will be added later by asking for contact info , incase of whatsapp will be added instantly a new user message)
6. **Bot Settings** (simple setting like which channels to be activated , adding credentials for that channel integration)

Bot setting page will have different content for different development plans ,
Because if client only choose website widget then there is no need of social media channels credentials)
7. **Help and notification** , (messaging portal between admin and client ,
It will be like whatsapp messenger
There will be two tabs
1. Notification (will get all notification like if paid for any month will get notification for

successful payment)

2. Help ..

8. Payment & subscription :

Client can

1. Pay due : client will see due payment and can pay by just clicking pay now
2. See analytics related payment
3. Manage subscription

9. Account Settings

Frontend - admin UI :

This is the interface from where i(my company) will manage all the clients and their chatbots ,

Pages in admin UI :

1. Dashboard

2. **Analytics** (overall analytics): no of clients , no of running projects , , total number of messages in different interval of time , and so on

3. **Payments** : payment related analytics and settings

financial analytics like earnings

Due payments , due having clients (,

4. **Users** :

All the registered clients(user for now) will be listed here ,

By clicking them admin can their profile info

(after new client registers admin will contact them and will help to choose a perfect plan for their requirement ,

As user selects any plan and pays initial development cost , then that client will be moved to running project)

5. **Running projects** :

As a new user is added to running project ,

Admin can see their profile info , choose plan , all the credentials for integration to different channels and all the setting related

For all the communication during development , admin can directly message to client from here which will be sent through customer service portal and user can that in help messenger portal

As the project is ready to demo the user can test it and as they are satisfied they will ask admin

about it

Then , admin will set it as finished ,

And it will send notification to pay the remaining development fee ,

Starting From that day user have to pay within 7 days,

As due is paid and user choose a subscription plan , that user will be moved to clients automatically (officially becomes client) ,

All the message sent before becoming client will not be included in to pay for them

Analytics will be refreshed

I mean, as a user becomes a client, it's all the data used to test the bot will be erased , like messages , contacts of customers , ...

6. **Clients** : Everything related any client (here client means who has a bot already built and has running subscription)
Analytics of that client
All the controls of that client and their bot ,

There will be list of all the clients ,

On clicking any client, a detailed info page of that client will open , which contains everything about the client , from its profile information to their analytics(as many analytics a client can see about their bot) , subscription type , due payments , ...

Activated channels , see their credentials for those channels (not their login credentials but the credentials they provided eg. whatsapp api ,)

And also can do whatever admin can do to a client ,

Suspending any account , sending message (which will send through customer service page) ,

7. **System setting** : setting for whole system
 1. Changing subscription plan : see and change all subscription plans by adding amount and description
 2. other settings
8. **Account setting**
9. **Customer service** : to respond customer queries asked through help portal

There is only one condition where admin can see any clients login credentials and can login into that account to examine their account ,

1. **In running project mode** : if any user is in running project mode , admin can see their login credentials , because during development bot is to be tested

After that running project is moved to clients category admin will not have access to login credentials of client

Clients will get max 7 days to pay for subscription renewal ,
I mean if a client has Npr 5k plan then start of new month they have to pay , they will get at max 7 days , after that they will be charged penalty of 5% of subscription fee per day ,

Database:

Database is very crucial part here because this application and the core n8n agent will communicate through this database ,

By adding and consuming each others added resources

For example :

the messages will be stored in message history table by n8n agent as it responds incoming messages and this backend will use those messages from table and show on client UI

Database structure will be hybrid for some types of data we will use shared table for all tenants (clients) using tenant_id ,

At the same time for some data separate table will be created for each tenant (client)

1. For vector storage each client will have separate table since each client will have separate KB and may need different configuration or features
2. In message history , we have to provide a table name in n8n , but while taking message context it use session_id , so we will use same table for all clients but session id will contain all three, client id + channel name + user id and when new message is added to table using js function we will extract client id , channel name , and user id and add

these into the same table in same row , so that while rendering in frontend channel wise message of each user can be shown

3. For other data think yourself and do which approach is better