Progress Report 2 - Group 4

Team: ConnectX https://github.com/Rwadsariya/ConnectX_4p02

Sprint 3 Duration: February 18 – March 6

Sprint 3 Goal:

The objective of Sprint 3 was to complete the front-end development of the application and begin laying the groundwork for the backend integration.

Sprint 3 Accomplishments:

Front-End Development:

- Completed full front-end implementation including:
  - Home Page
  - Pricing Page
  - Features Page
  - About Page
  - Dashboard Page
- Interface aligned with the Figma designs and offers full navigational flow and functionality.

Back-End Setup:

- Began backend setup in preparation for full implementation in Sprint 4.

Feature Development:

- Integrated automated email system for welcoming new users.

Sprint 4 Duration: March 6 – March 20

Sprint 4 Goal:

The objective of Sprint 4 was to complete the back-end systems of the application, including logic, integrations, and automation.

Sprint 4 Accomplishments:

Back-End Development:

- Completed webhook implementation for automated user actions.
- Connected and configured OpenAI API for automated responses.
- Integrated Stripe for handling payments.
- Finalized the automation system for both free and paid users.

Dashboard:

- Finalized and implemented the full dashboard view with working backend hooks.
- Created all required components for functionality and user interface.

Challenges Faced:

- Setting up the Instagram API and webhooks required extensive research and debugging due to the platform’s complexity and restrictive documentation. This is because to get an instagram api key, you first need to create a page on facebook and link that with your development profile. Then you have to link that with your instagram page which then allows you to write scripts for it.
- As with previous sprints, coordinating in-person meetings was difficult due to varied schedules, but the team successfully conducted all required scrums, sprint reviews, and retrospectives in person.

Next Steps (Sprint 5 Plan):

In the next sprint, the team will focus on:

- Writing JUnit tests for backend systems.
- Connecting the system to the Instagram API for automated DMs and comment replies.
- Performing UI refinements and polish based on feedback.
- Creating helper functions to handle niche cases.

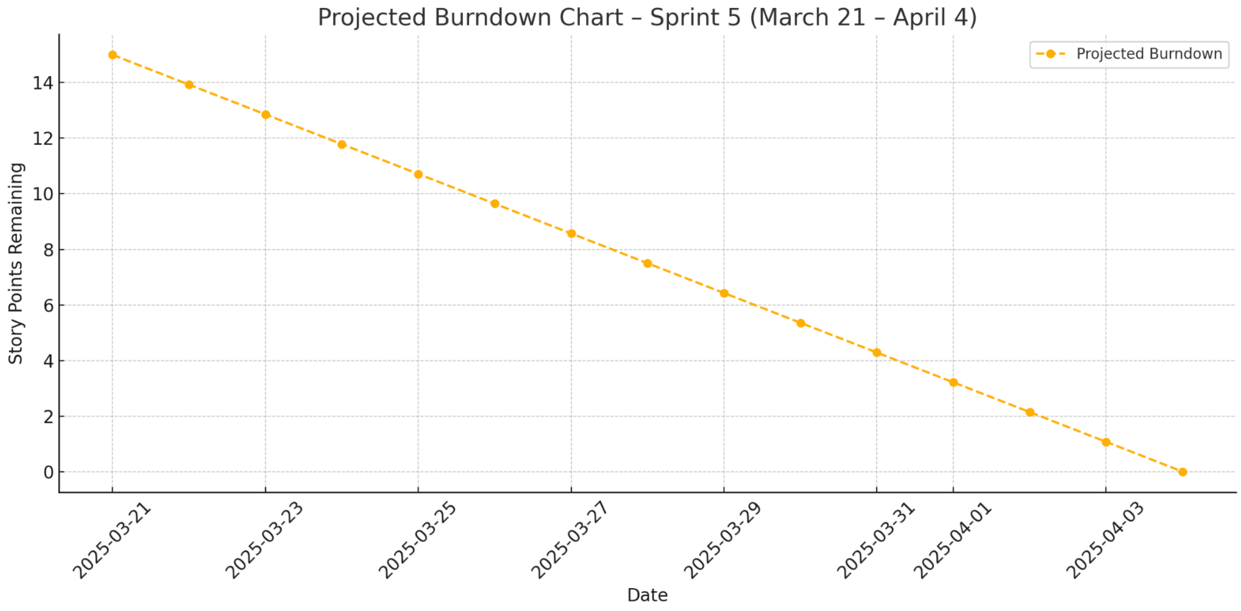
Individual Contributions:

- **Basil Syed** – Created webhook and connected OpenAI API to system backend for intelligent responses.
- **Neel Khambhata** – Built core components on both front-end and backend sides to support UI and logic workflows.
- **Romin Wadsariya** – Developed automation for both free and paid user flows, handling post comments and DM logic.
- **Mann Patel & Kashish Patel** – Fully designed and implemented the entire front-end interface from scratch.
- **Nitish Bhardwaj** – Set up the automated email pipeline and implemented Stripe payments integration.

Meetings:

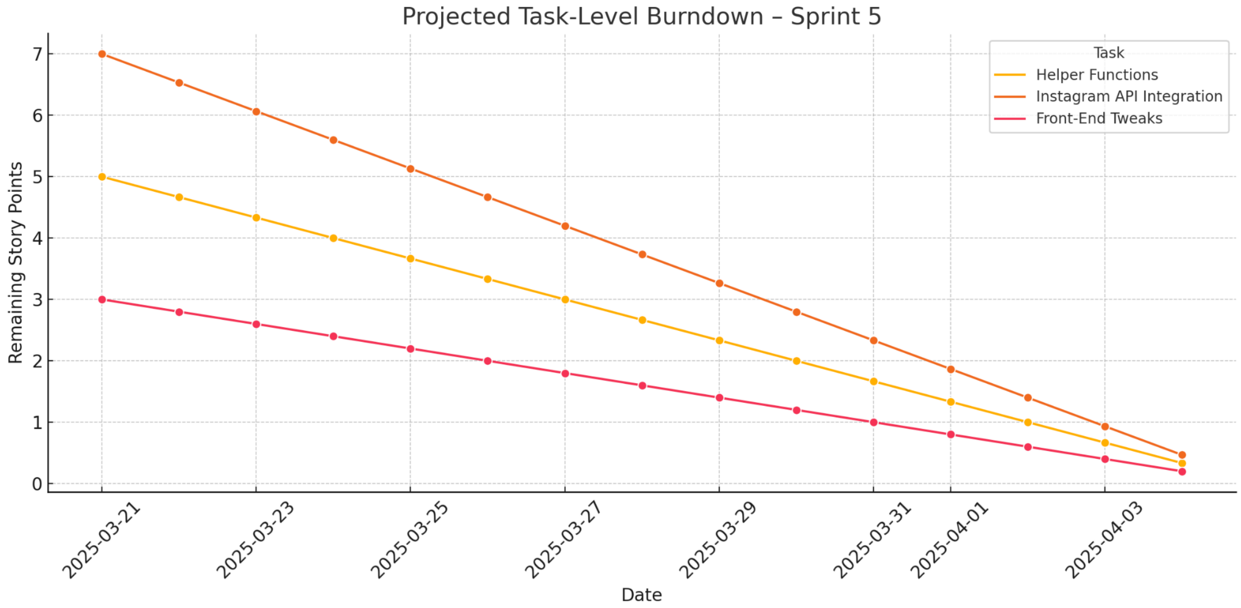
Held all meetings in person for sprints 3 and 4, including one scrum every week, one sprint review at the end of each sprint and one sprint retrospective to realize work done and what's left.

Burn down charts for work left:



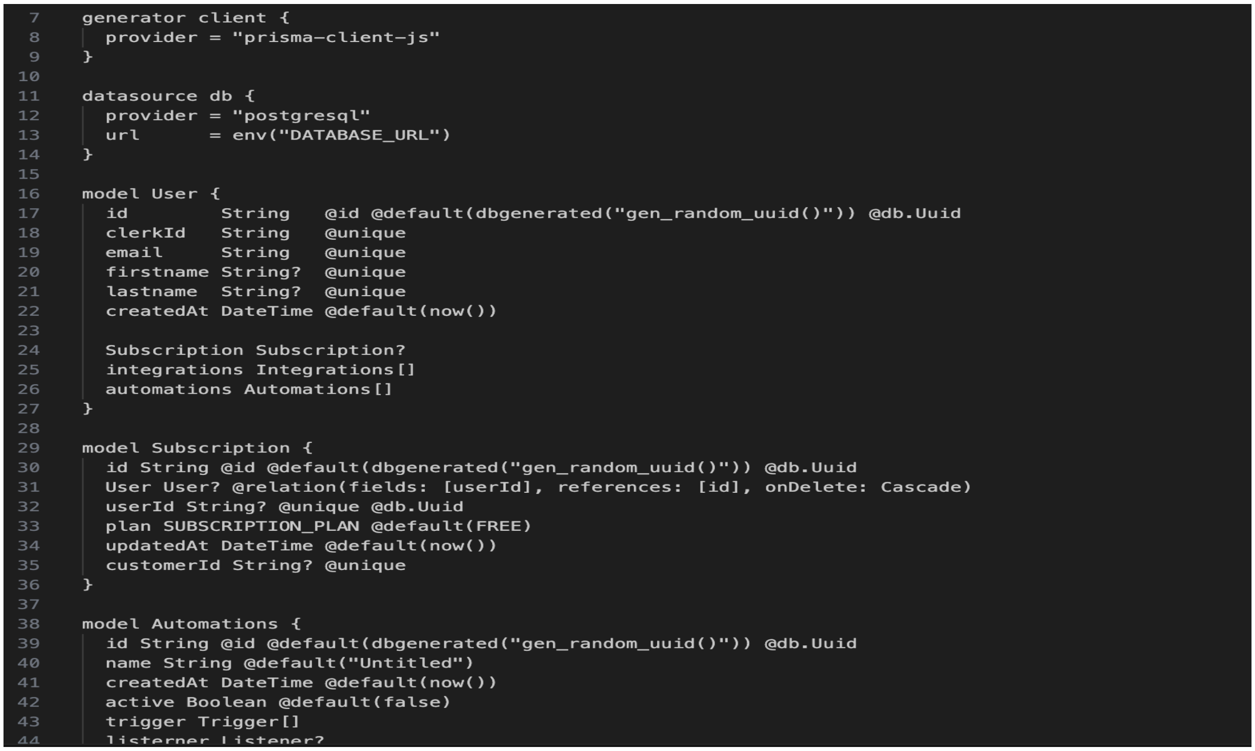
Sprint 5 Task Assignment:

Task	Story Points	Team Members
Helper Functions	5	Neel, Nitish
Instagram API Integration	7	Romin, Basil
Front-End Tweaks	3	Mann, Kashish



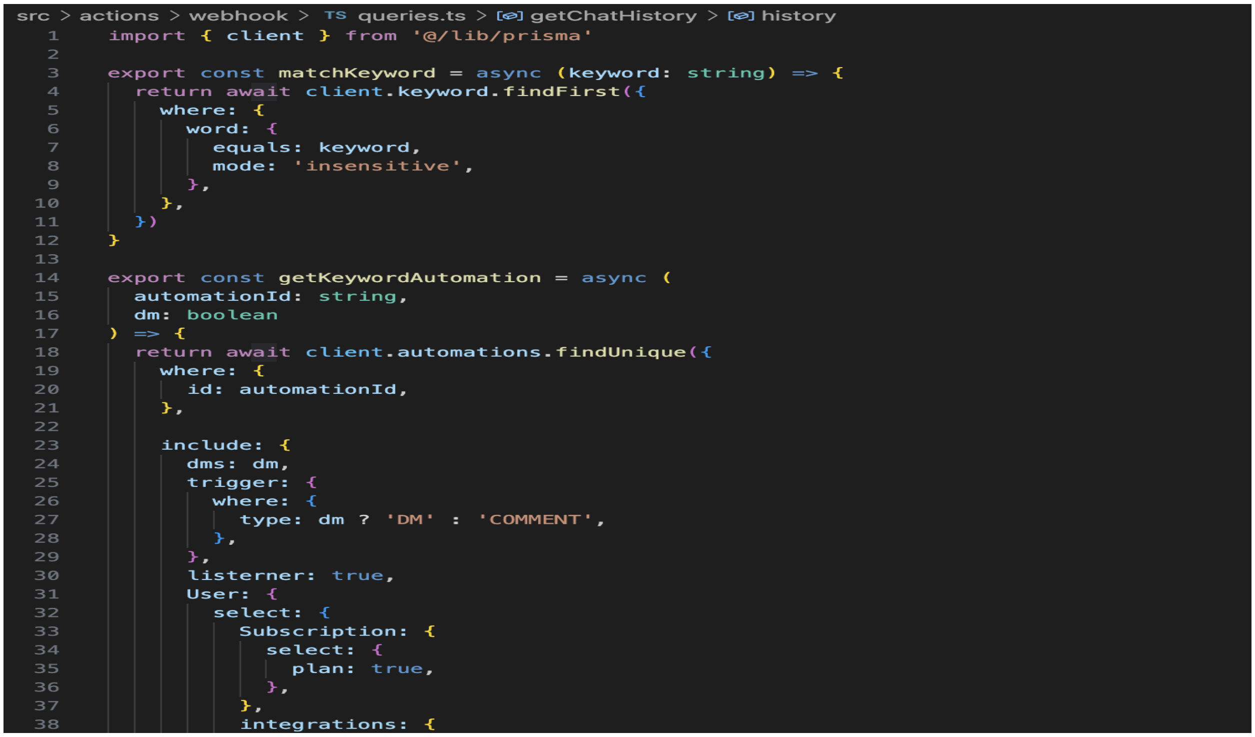
Images:

Prisma file for database creation:



```
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id        String   @id @default(dbgenerated("gen_random_uuid()")) @db.Uuid
  clerkId   String   @unique
  email     String   @unique
  firstname String?  @unique
  lastname  String?  @unique
  createdAt DateTime @default(now())

  Subscription Subscription?
  integrations Integrations[]
  automations Automations[]
}

model Subscription {
  id String @id @default(dbgenerated("gen_random_uuid()")) @db.Uuid
  User User? @relation(fields: [userId], references: [id], onDelete: Cascade)
  userId String? @unique @db.Uuid
  plan SUBSCRIPTION_PLAN @default(FREE)
  updatedAt DateTime @default(now())
  customerId String? @unique
}

model Automations {
  id String @id @default(dbgenerated("gen_random_uuid()")) @db.Uuid
  name String @default("Untitled")
  createdAt DateTime @default(now())
  active Boolean @default(false)
  trigger Trigger[]
  listerner Listener?
```

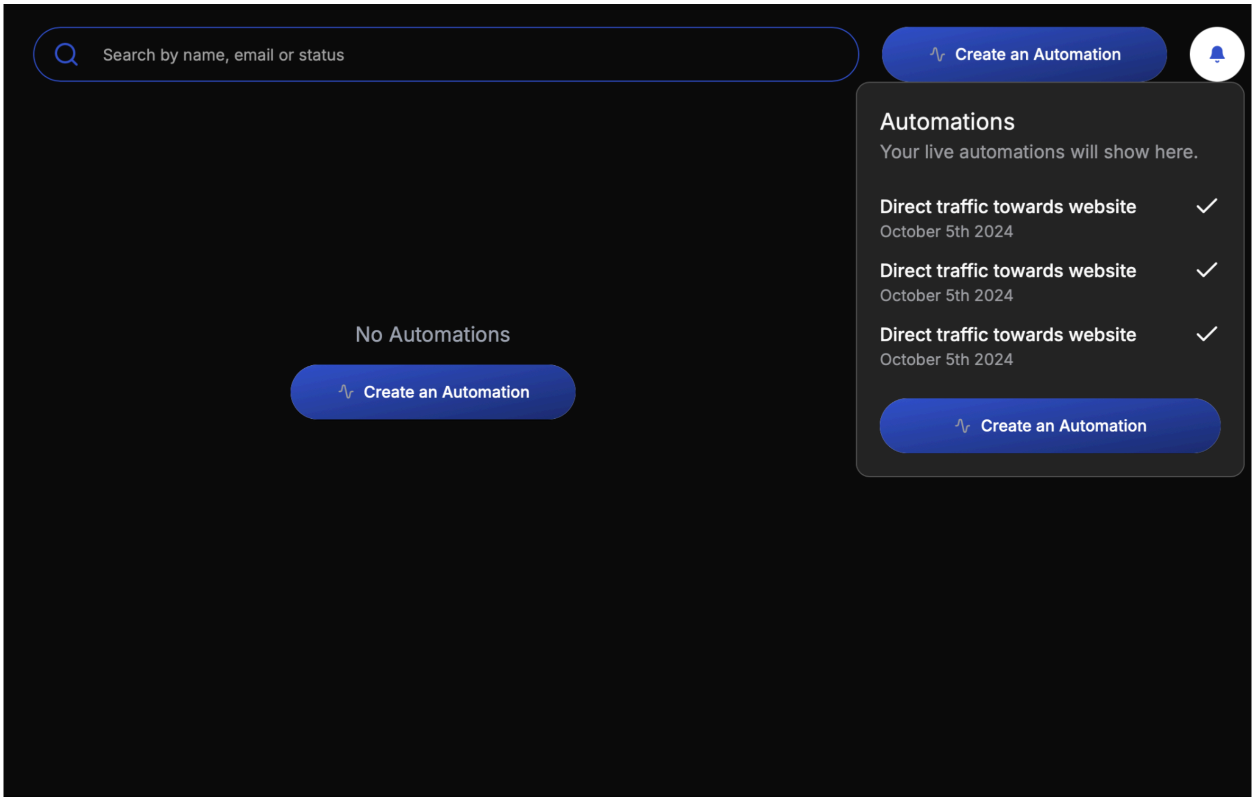
Webhook actions:



src > actions > webhook > TS queries.ts > getChatHistory > history

```
import { client } from '@/lib/prisma'

export const matchKeyword = async (keyword: string) => {
  return await client.keyword.findFirst({
    where: {
      word: {
        equals: keyword,
        mode: 'insensitive',
      },
    },
  })
}

export const getKeywordAutomation = async (
  automationId: string,
  dm: boolean
) => {
  return await client.automations.findUnique({
    where: {
      id: automationId,
    },

    include: {
      dms: dm,
      trigger: {
        where: {
          type: dm ? 'DM' : 'COMMENT',
        },
      },
      listerner: true,
      User: {
        select: {
          Subscription: {
            select: {
              plan: true,
            },
          },
          integrations: {
```

Automations dashboard:



No Automations

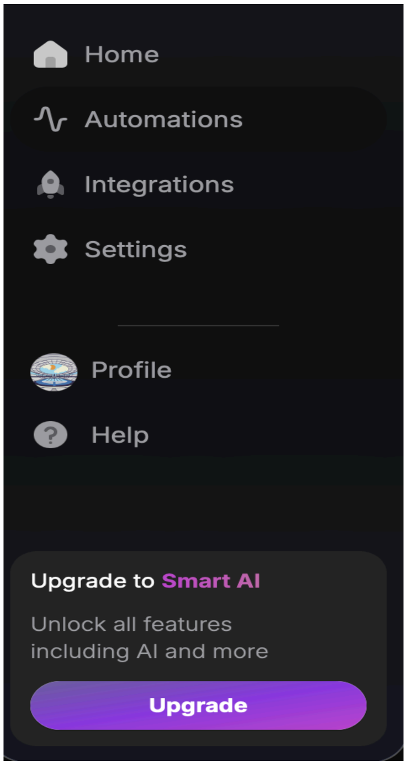
Automations

Your live automations will show here.

Direct traffic towards website ✓
October 5th 2024

Direct traffic towards website ✓
October 5th 2024

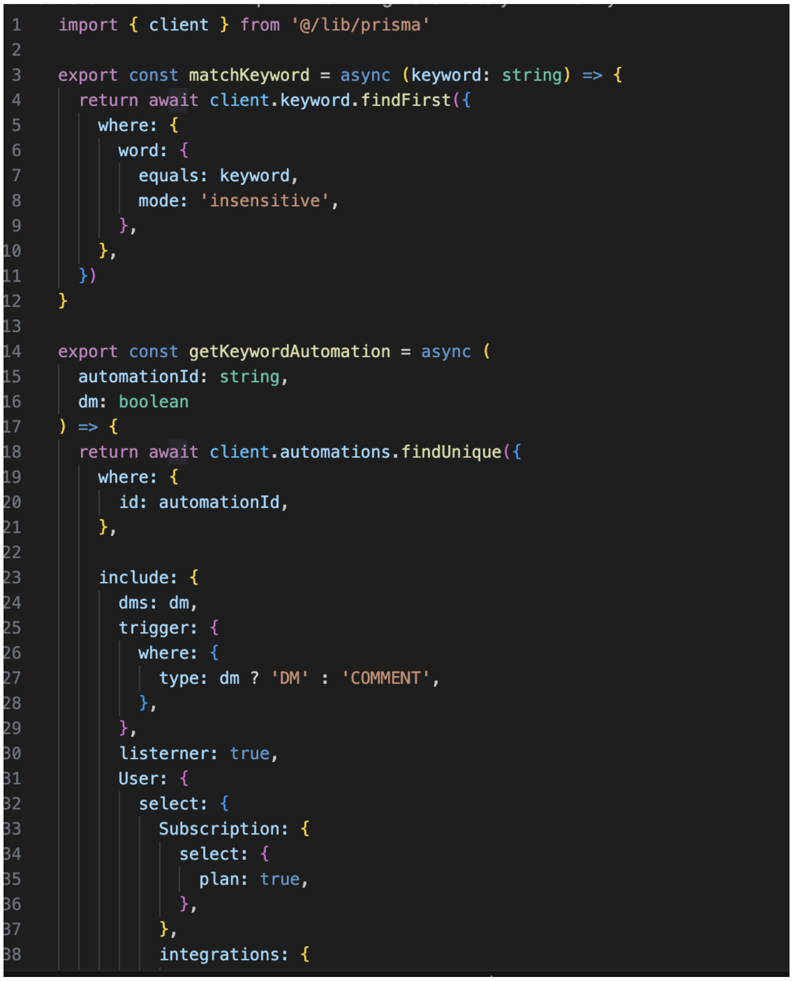
Direct traffic towards website ✓
October 5th 2024



Upgrade to Smart AI

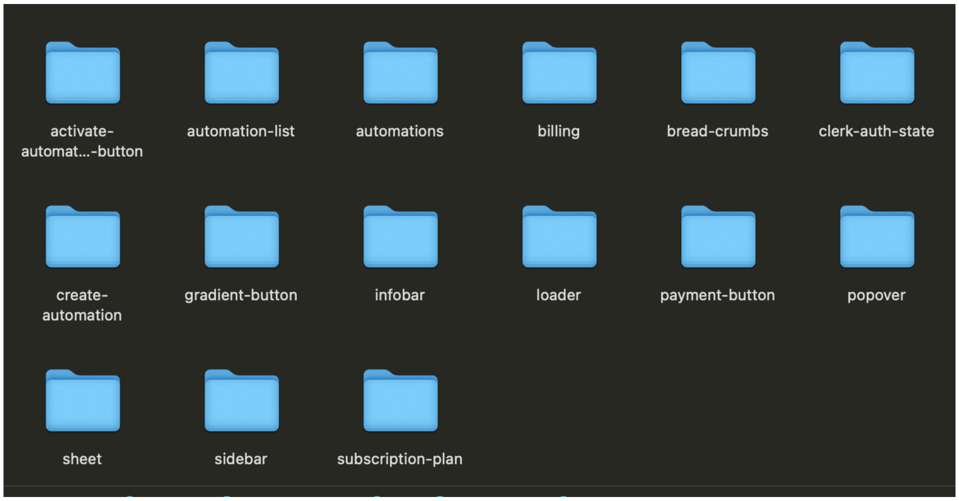
Unlock all features including AI and more

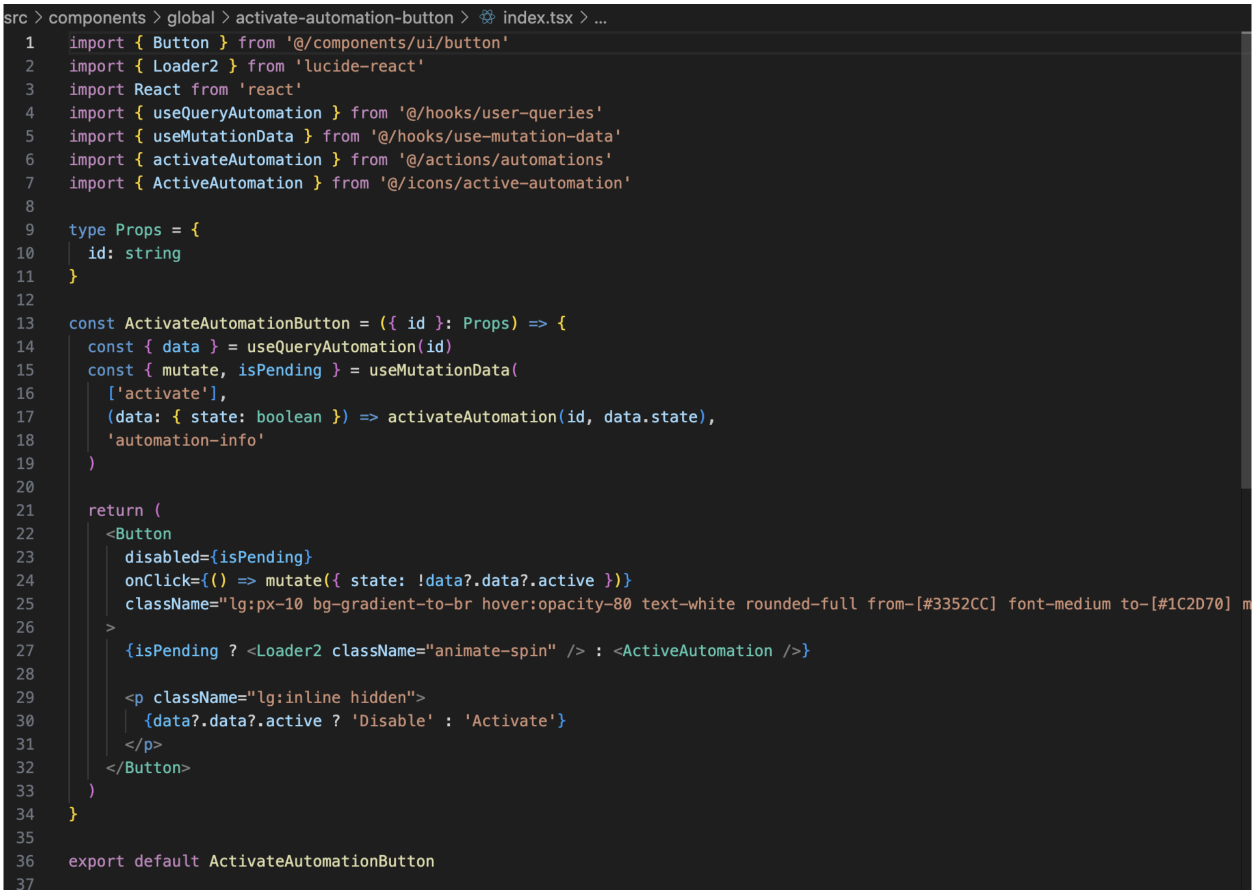
Webhook cases:



```
import { client } from '@/lib/prisma'

export const matchKeyword = async (keyword: string) => {
  return await client.keyword.findFirst({
    where: {
      word: {
        equals: keyword,
        mode: 'insensitive',
      },
    },
  })
}

export const getKeywordAutomation = async (
  automationId: string,
  dm: boolean
) => {
  return await client.automations.findUnique({
    where: {
      id: automationId,
    },

    include: {
      dms: dm,
      trigger: {
        where: {
          type: dm ? 'DM' : 'COMMENT',
        },
      },
      listerner: true,
      User: {
        select: {
          Subscription: {
            select: {
              plan: true,
            },
          },
          integrations: {
```

Components:

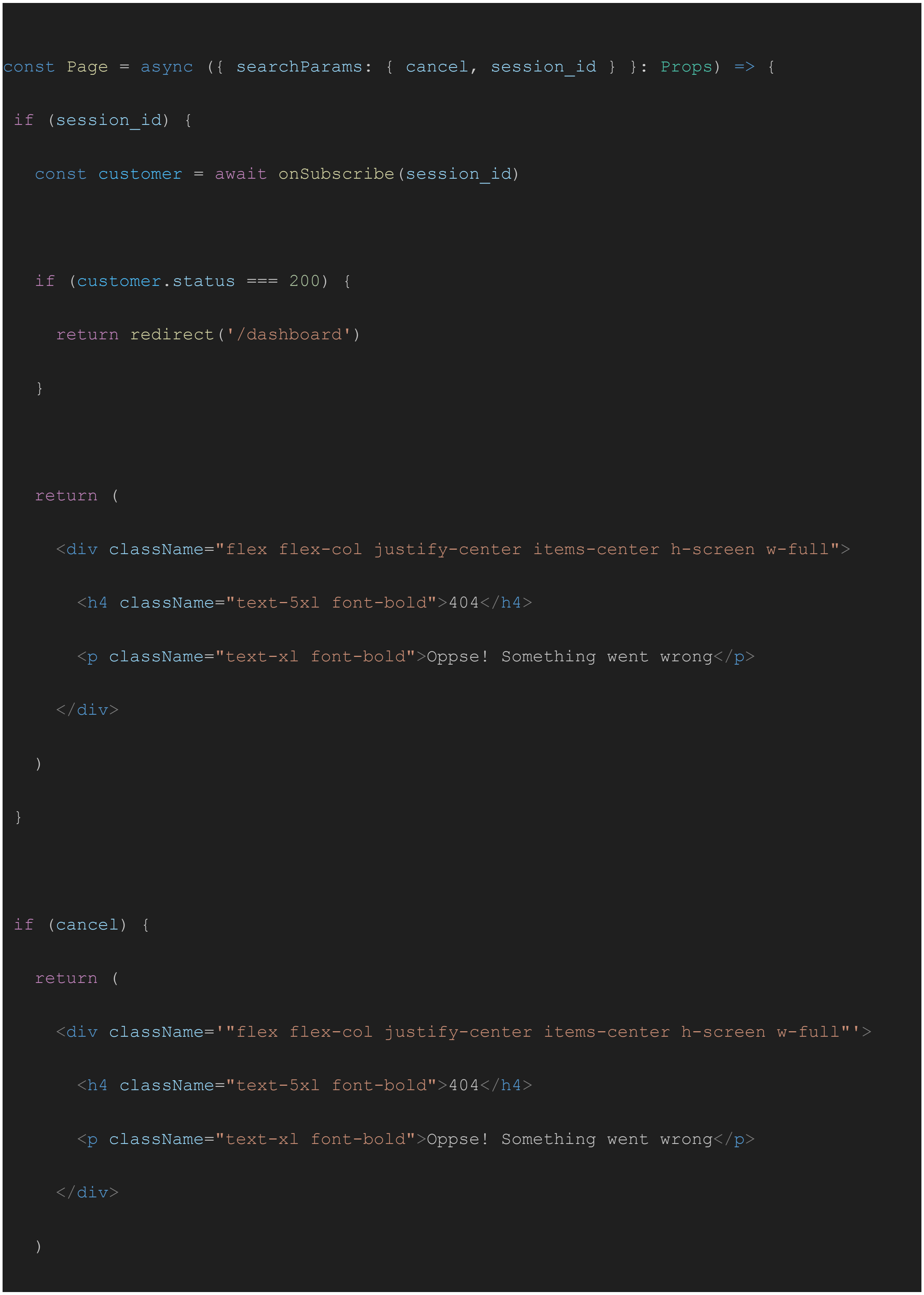



src > components > global > activate-automation-button > index.tsx > ...

```tsx
import { Button } from '@/components/ui/button'
import { Loader2 } from 'lucide-react'
import React from 'react'
import { useQueryAutomation } from '@/hooks/user-queries'
import { useMutationData } from '@/hooks/use-mutation-data'
import { activateAutomation } from '@/actions/automations'
import { ActiveAutomation } from '@/icons/active-automation'

type Props = {
  id: string
}

const ActivateAutomationButton = ({ id }: Props) => {
  const { data } = useQueryAutomation(id)
  const { mutate, isPending } = useMutationData(
    ['activate'],
    (data: { state: boolean }) => activateAutomation(id, data.state),
    'automation-info'
  )

  return (
    <Button
      disabled={isPending}
      onClick={() => mutate({ state: !data?.data?.active })}
      className="lg:px-10 bg-gradient-to-br hover:opacity-80 text-white rounded-full from-[#3352CC] font-medium to-[#1C2D70] m
    >
      {isPending ? <Loader2 className="animate-spin" /> : <ActiveAutomation />}

      <p className="lg:inline hidden">
        {data?.data?.active ? 'Disable' : 'Activate'}
      </p>
    </Button>
  )
}

export default ActivateAutomationButton
```

Stripe Payment Page:

```tsx
import { onSubscribe } from '@/actions/user'

import { redirect } from 'next/navigation'

import React from 'react'



type Props = {

 searchParams: {

   session_id?: string

   cancel?: boolean

 }

}
```



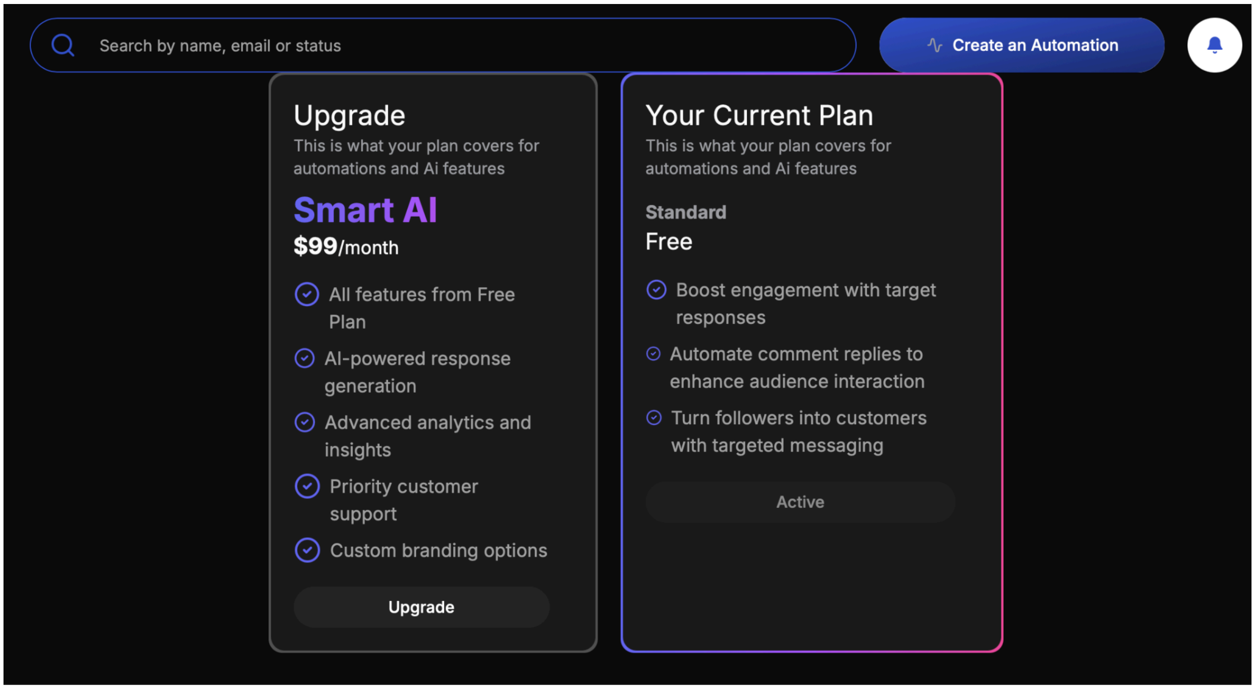
```
const Page = async ({ searchParams: { cancel, session_id } }: Props) => {
 if (session_id) {
   const customer = await onSubscribe(session_id)

   if (customer.status === 200) {
     return redirect('/dashboard')
   }

   return (
     <div className="flex flex-col justify-center items-center h-screen w-full">
       <h4 className="text-5xl font-bold">404</h4>
       <p className="text-xl font-bold">Oppse! Something went wrong</p>
     </div>
   )
 }

 if (cancel) {
   return (
     <div className='"flex flex-col justify-center items-center h-screen w-full"'>
       <h4 className="text-5xl font-bold">404</h4>
       <p className="text-xl font-bold">Oppse! Something went wrong</p>
     </div>
   )
 }

}



export default Page
```

Front page:

```
import { Button } from '@/components/ui/button'
import {

 Card,

 CardContent,

 CardDescription,

 CardFooter,

 CardHeader,

 CardTitle,

} from '@/components/ui/card'

import { CheckCircle, MenuIcon } from 'lucide-react'

import Image from 'next/image'

import Link from 'next/link'



export default function Home() {

 const plans = [

   {

     name: 'Free Plan',
     description: 'Perfect for getting started',

     price: '$0',

     features: [

       'Boost engagement with target responses',

       'Automate comment replies to enhance audience interaction',

       'Turn followers into customers with targeted messaging',

     ],

     cta: 'Get Started',

   },

   {

     name: 'Smart AI Plan',

     description: 'Advanced features for power users',

     price: '$99',

     features: [

       'All features from Free Plan',

       'AI-powered response generation',

       'Advanced analytics and insights',

       'Priority customer support',

       'Custom branding options',

     ],

     cta: 'Upgrade Now',

   },

 ]

 return (
   <main>

     <section className="relative bg-gradient-to-b from-slate-900 via-blue-900 to-bg">

       <div className="absolute inset-0 bg-[linear-gradient(to_right,#0f172a_1px,transparent_1px),linear-gradient(to_bottom,#0f172a_1px,transparent_1px)] bg-[size:4rem_4rem] [mask-image:radial-gradient(ellipse_60%_50%_at_50%_0%,#000_70%,transparent_110%)]" />

       <div className="relative">

         <div className="container px-4 py-8">

           <div className="flex items-center justify-between">

             <div className="flex items-center gap-2">

               <div className="h-8 w-8 rounded-lg bg-white flex items-center justify-center font-bold">

                 li

               </div>

               <span className="text-xl font-semibold text-primary-foreground">

                 Slide

               </span>

             </div>

             <nav className="hidden space-x-6 text-sm text-blue-200 md:block">

               <Link href="#features">Features</Link>

               <Link href="#pricing">Pricing</Link>

               <Link href="#about">About</Link>

             </nav>

             <Button className="bg-black text-primary">

               <Link href="/dashboard">Login</Link>

             </Button>
           </div>


           <div className="mx-auto mt-16 max-w-3xl text-center">
             <h1 className="text-4xl font-bold leading-tight tracking-tighter text-white sm:text-5xl md:text-6xl lg:text-7xl">
               Transform Your Instagram Engagement with Slide
             </h1>


             <p className="mt-6 text-lg text-blue-200">
               Slide revolutionizes how you connect with your audience on
               Instagram. Automate responses and boost engagement effortlessly,
               turning interactions into valuable business opportunities.
             </p>


             <div className="mt-8 flex justify-center gap-4">
               <Button
                 size="lg"
                 className="bg-blue-600 text-white hover:bg-blue-700"
               >
                 Get Started
               </Button>
               <Button
                 size="lg"
                 variant="outline"
```

```
                className="border-blue-400  hover:bg-blue-900/50"
              >
                Learn More
              </Button>
            </div>
          </div>
          <div className="relative h-40 md:h-80 w-full  mt-10">
            <Image
              src="/Ig-creators.png"
              alt="Community member"
              fill
              className="object-cover"
            />
          </div>
        </div>
      </div>
    </section>
    <section className="container w-full py-12 md:py-24 lg:py-32 bg-background">
      <div className="container px-4 md:px-6">
        <div className="flex flex-col items-center justify-center space-y-4 text-center">
          <h2 className="text-3xl font-bold tracking-tighter sm:text-5xl">
            Choose Your Plan
          </h2>
          <p className="max-w-[900px] text-muted-foreground">
            Select the perfect plan to boost your Instagram engagement
          </p>
        </div>
        <div className="grid grid-cols-1 gap-6 mt-8 md:grid-cols-2 md:gap-8">
          {plans.map((plan, index) => (
            <Card
              key={index}
              className="flex flex-col justify-between"
            >
              <CardHeader>
                <CardTitle>{plan.name}</CardTitle>
                <CardDescription>{plan.description}</CardDescription>
              </CardHeader>
              <CardContent className="grid gap-4">
                <div className="text-4xl font-bold">
                  {plan.price}
                  <span className="text-lg font-normal text-muted-foreground">
                    /month
                  </span>
                </div>
                <ul className="space-y-2">
                  {plan.features.map((feature, i) => (
                    <li
```

```
                       key={i}
                       className="flex items-center"
                     >
                       <CheckCircle className="mr-2 h-4 w-4 text-primary" />
                       <span className="text-sm text-muted-foreground">
                         {feature}
                       </span>
                     </li>
                   ))}
                 </ul>
               </CardContent>
               <CardFooter>
                 <Button className="w-full">{plan.cta}</Button>
               </CardFooter>
             </Card>
           ))}
         </div>
       </div>
     </section>
   </main>
 )
}
```



Upgrade

This is what your plan covers for automations and Ai features

Smart AI

$99/month

- ✓ All features from Free Plan
- ✓ AI-powered response generation
- ✓ Advanced analytics and insights
- ✓ Priority customer support
- ✓ Custom branding options

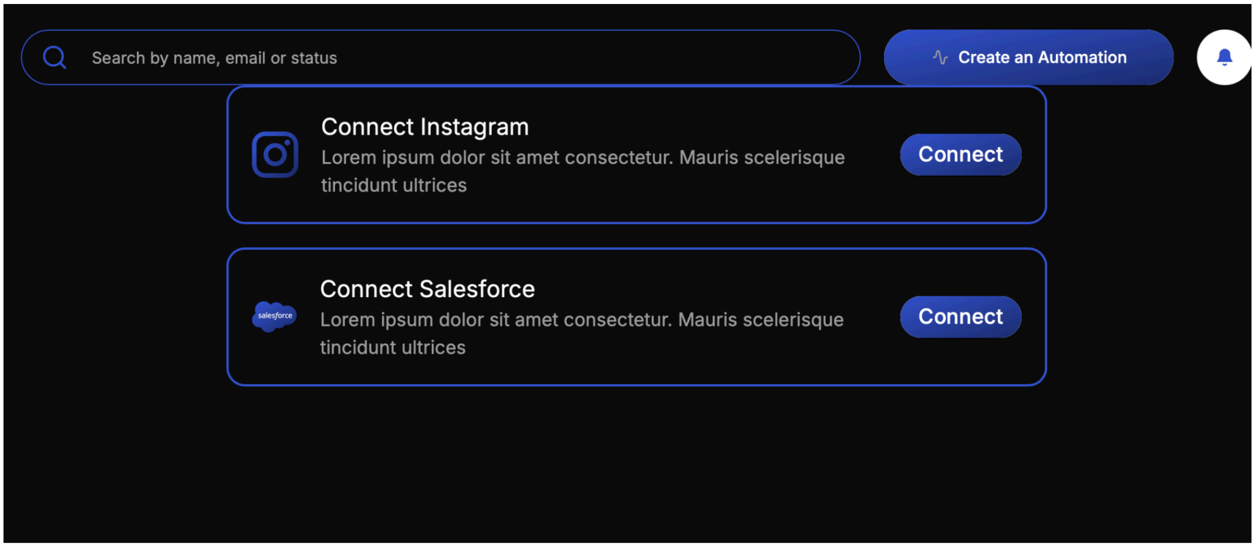
Your Current Plan

This is what your plan covers for automations and Ai features

Standard

Free

- ✓ Boost engagement with target responses
- ✓ Automate comment replies to enhance audience interaction
- ✓ Turn followers into customers with targeted messaging



Connect Instagram

Lorem ipsum dolor sit amet consectetur. Mauris scelerisque tincidunt ultrices

Connect Salesforce

Lorem ipsum dolor sit amet consectetur. Mauris scelerisque tincidunt ultrices