import React from 'react';

import {

  InfoSec,

  InfoRow,

  InfoColumn,

  TextWrapper,

  TopLine,

  Heading,

  Subtitle,

  ImgWrapper,

  Img,

  Container,

  StyledButton

} from './InfoSection.elements';

import { Link } from 'react-router-dom';

interface InfoSectionProps {

  primary?: boolean;

  lightBg?: boolean;

  topLine?: string;

  lightTopLine?: boolean;

  lightText?: boolean;

  lightTextDesc?: boolean;

  headline?: string;

  description?: string;

  buttonLabel?: string;

  img?: string;

  alt?: string;

  imgStart?: boolean;

  start?: boolean;

}

const InfoSection: React.FC<InfoSectionProps> = ({

  primary,

  lightBg,

  topLine,

  lightTopLine,

  lightText,

  lightTextDesc,

  headline,

  description,

  buttonLabel,

  img,

  alt,

  imgStart,

  start

}) => {

  return (

    <InfoSec lightBg={lightBg}>

      <Container>

        <InfoRow imgStart={imgStart}>

          <InfoColumn>

            <TextWrapper>

              <TopLine lightTopLine={lightTopLine}>{topLine}</TopLine>

              <Heading lightText={lightText}>{headline}</Heading>

              <Subtitle lightTextDesc={lightTextDesc}>{description}</Subtitle>

              <Link to="/sign-up">

                <StyledButton primary={primary}>{buttonLabel}</StyledButton>

              </Link>

            </TextWrapper>

          </InfoColumn>

          <InfoColumn>

            <ImgWrapper start={start}>

              <Img src={img} alt={alt} />

            </ImgWrapper>

          </InfoColumn>

        </InfoRow>

      </Container>

    </InfoSec>

  );

};

export default InfoSection;

infosection.tsx file code   
  
import styled from 'styled-components';

import { Link } from 'react-router-dom';

interface NavMenuProps {

  click: boolean;

}

export const Nav = styled.nav`

  background: #101522;

  height: 80px;

  display: flex;

  justify-content: center;

  align-items: center;

  font-size: 1rem;

  position: sticky;

  top: 0;

  z-index: 10;

  @media screen and (max-width: 960px) {

    transition: 0.8s all ease;

  }

`;

export const NavbarContainer = styled.div`

  display: flex;

  justify-content: space-between;

  height: 80px;

  z-index: 1;

  width: 100%;

  padding: 0 24px;

  max-width: 1100px;

`;

export const NavLogo = styled(Link)`

  color: #fff;

  justify-self: flex-start;

  cursor: pointer;

  font-size: 1.5rem;

  display: flex;

  align-items: center;

  margin-left: 24px;

  font-weight: bold;

  text-decoration: none;

`;

export const MobileIcon = styled.div`

  display: none;

  @media screen and (max-width: 768px) {

    display: block;

    position: absolute;

    top: 0;

    right: 0;

    transform: translate(-100%, 60%);

    font-size: 1.8rem;

    cursor: pointer;

    color: #fff;

  }

`;

export const NavMenu = styled.ul<NavMenuProps>`

  display: flex;

  align-items: center;

  list-style: none;

  text-align: center;

  margin-right: -22px;

  @media screen and (max-width: 768px) {

    display: ${({ click }) => (click ? 'flex' : 'none')};

    flex-direction: column;

    width: 100%;

    position: absolute;

    top: 80px;

    left: 0;

    background: #101522;

    transition: all 0.3s ease;

  }

`;

export const NavItem = styled.li`

  height: 80px;

`;

export const NavLinks = styled(Link)`

  color: #fff;

  display: flex;

  align-items: center;

  text-decoration: none;

  padding: 0 1rem;

  height: 100%;

  cursor: pointer;

  &:hover {

    border-bottom: 3px solid #01bf71;

    color: #01bf71;

  }

`;

export const ProfileWrapper = styled.div`

  display: flex;

  align-items: center;

  gap: 8px;

  color: #fff;

  font-weight: 500;

  font-size: 0.95rem;

  padding: 0 1rem;

`;

export const UserName = styled.span`

  color: #fff;

  font-size: 0.95rem;

  overflow: hidden;

  text-overflow: ellipsis;

  white-space: nowrap;

  max-width: 120px;

  @media screen and (max-width: 768px) {

    max-width: 100px;

  }

`;

export const LogoutButton = styled.button`

  background: transparent;

  color: #ff6b6b;

  border: 1px solid #ff6b6b;

  border-radius: 5px;

  padding: 6px 12px;

  margin-left: 8px;

  cursor: pointer;

  transition: all 0.2s ease;

  &:hover {

    background: #ff6b6b;

    color: #fff;

  }

  @media screen and (max-width: 768px) {

    margin-top: 12px;

  }

`;

Navbar.elements.tsx file code  
  
import React from "react";

import { Link, useLocation, useNavigate } from "react-router-dom";

import styled from "styled-components";

import { useAuth } from "@/hooks/useAuth";

const Navbar: React.FC = () => {

  const location = useLocation();

  const navigate = useNavigate();

  const { user, logout } = useAuth();

  const handleLogout = () => {

    logout();

    navigate("/login");

  };

  return (

    <Nav>

      <Logo to="/">ATS Score</Logo>

      <NavLinks>

        {user ? (

          <>

            <StyledLink to="/dashboard" $active={location.pathname === "/dashboard"}>

              Dashboard

            </StyledLink>

            <StyledLink to="/history" $active={location.pathname === "/history"}>

              History

            </StyledLink>

          </>

        ) : null}

      </NavLinks>

      {user ? (

        <ProfileSection>

          <Avatar

            src={user.avatarUrl || `https://ui-avatars.com/api/?name=${user.name}`}

            alt="Profile"

          />

          <UserName>{user.name}</UserName>

          <LogoutButton onClick={handleLogout}>Logout</LogoutButton>

        </ProfileSection>

      ) : (

        <NavLinks>

          <StyledLink to="/login" $active={location.pathname === "/login"}>

            Login

          </StyledLink>

          <StyledLink to="/register" $active={location.pathname === "/register"}>

            Register

          </StyledLink>

        </NavLinks>

      )}

    </Nav>

  );

};

export default Navbar;

// ---------- Styled Components ----------

const Nav = styled.nav`

  height: 64px;

  padding: 0 2rem;

  background-color: #0f172a;

  color: white;

  display: flex;

  align-items: center;

  justify-content: space-between;

  border-bottom: 1px solid #1e293b;

`;

const Logo = styled(Link)`

  font-size: 1.5rem;

  font-weight: bold;

  text-decoration: none;

  color: white;

`;

const NavLinks = styled.div`

  display: flex;

  gap: 1rem;

  @media (max-width: 600px) {

    display: none;

  }

`;

const StyledLink = styled(Link)<{ $active?: boolean }>`

  color: ${({ $active }) => ($active ? "#38bdf8" : "white")};

  font-weight: 500;

  text-decoration: none;

  border-bottom: ${({ $active }) => ($active ? "2px solid #38bdf8" : "none")};

  padding-bottom: 2px;

  &:hover {

    color: #38bdf8;

  }

`;

const ProfileSection = styled.div`

  display: flex;

  align-items: center;

  gap: 0.75rem;

`;

const Avatar = styled.img`

  width: 32px;

  height: 32px;

  border-radius: 50%;

  object-fit: cover;

`;

const UserName = styled.span`

  font-weight: 500;

`;

const LogoutButton = styled.button`

  padding: 6px 12px;

  background-color: #1e293b;

  border: none;

  border-radius: 4px;

  color: white;

  font-size: 0.85rem;

  cursor: pointer;

  &:hover {

    background-color: #334155;

  }

`;

Navbar.tsx file code

import React from 'react';

import styled from 'styled-components';

const Header = styled.div`

  display: flex;

  flex-direction: column;

  gap: 4px;

`;

const Heading = styled.h1`

  font-size: 2rem;

  color: #ffffff;

`;

const SubHeading = styled.p`

  font-size: 1rem;

  color: #94a3b8;

`;

const GreetingHeader: React.FC = () => {

  const today = new Date().toLocaleDateString();

  return (

    <Header>

      <Heading>Welcome, Sanjay 👋</Heading>

      <SubHeading>{today}</SubHeading>

    </Header>

  );

};

export default GreetingHeader;

GreetingHeader.tsx file code

import { useAuth } from '@/hooks/useAuth';

import { Navigate } from 'react-router-dom';

interface ProtectedRouteProps {

  children: React.ReactNode;

}

export function ProtectedRoute({ children }: ProtectedRouteProps) {

  const { user, loading } = useAuth();

  if (loading) {

    return (

      <div className="min-h-screen flex items-center justify-center">

        <div className="animate-spin rounded-full h-32 w-32 border-b-2 border-primary"></div>

      </div>

    );

  }

  if (!user) {

    return <Navigate to="/login" replace />;

  }

  return <>{children}</>;

}

export default ProtectedRoute;

ProtectedRoute.tsx file code   
  
import React from 'react';

import styled from 'styled-components';

import { motion } from 'framer-motion';

interface ResumeCardProps {

  title: string;

  date: string;

  score: number;

}

const Card = styled(motion.div)`

  background-color: #1e293b;

  border-radius: 12px;

  padding: 1.5rem;

  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.3);

  display: flex;

  flex-direction: column;

  gap: 1rem;

`;

const Title = styled.h3`

  font-size: 1.1rem;

  font-weight: 600;

  color: white;

`;

const DateText = styled.span`

  font-size: 0.9rem;

  color: #94a3b8;

`;

const ProgressBar = styled.div`

  height: 12px;

  border-radius: 8px;

  overflow: hidden;

  background-color: #334155;

`;

const Progress = styled.div<{ $score: number }>`

  height: 100%;

  width: ${({ $score }) => `${$score}%`};

  background-color: ${({ $score }) =>

    $score >= 80 ? '#22c55e' :

    $score >= 60 ? '#facc15' :

    '#f43f5e'};

`;

const Score = styled.div`

  font-size: 1rem;

  font-weight: 500;

  color: #e2e8f0;

`;

const Button = styled.button`

  margin-top: auto;

  background-color: transparent;

  border: 1px solid #94a3b8;

  color: #e2e8f0;

  padding: 0.5rem 1rem;

  font-size: 0.9rem;

  border-radius: 6px;

  cursor: pointer;

  &:hover {

    background-color: #334155;

  }

`;

const ResumeCard: React.FC<ResumeCardProps> = ({ title, date, score }) => {

  return (

    <Card

      initial={{ opacity: 0, y: 20 }}

      animate={{ opacity: 1, y: 0 }}

      transition={{ duration: 0.4 }}

    >

      <Title>{title}</Title>

      <DateText>Uploaded: {date}</DateText>

      <Score>ATS Score: {score}%</Score>

      <ProgressBar>

        <Progress $score={score} />

      </ProgressBar>

      <Button>View Details</Button>

    </Card>

  );

};

export default ResumeCard;

ResumeCard.tsx file code

import React from 'react';

import styled from 'styled-components';

import { motion } from 'framer-motion';

const Button = styled(motion.button)`

  padding: 10px 20px;

  background-color: #2563eb;

  color: #ffffff;

  border: none;

  border-radius: 8px;

  cursor: pointer;

  font-weight: 600;

`;

const UploadButton: React.FC = () => {

  return (

    <Button

      whileHover={{ scale: 1.05 }}

      whileTap={{ scale: 0.95 }}

    >

      Upload Resume

    </Button>

  );

};

export default UploadButton;

UploadButton.tsx file code

// src/contexts/UserContext.tsx

import React, { createContext, useContext, useState, useEffect } from 'react';

type User = {

  name: string;

  email: string;

  avatarUrl?: string;

};

type AuthContextType = {

  user: User | null;

  login: (userData: User) => void;

  logout: () => void;

};

const AuthContext = createContext<AuthContextType | undefined>(undefined);

export const UserProvider: React.FC<{ children: React.ReactNode }> = ({ children }) => {

  const [user, setUser] = useState<User | null>(null);

  // Load user from localStorage on mount

  useEffect(() => {

    const storedUser = localStorage.getItem('user');

    if (storedUser) {

      setUser(JSON.parse(storedUser));

    }

  }, []);

  const login = (userData: User) => {

    setUser(userData);

    localStorage.setItem('user', JSON.stringify(userData));

  };

  const logout = () => {

    setUser(null);

    localStorage.removeItem('user');

  };

  return (

    <AuthContext.Provider value={{ user, login, logout }}>

      {children}

    </AuthContext.Provider>

  );

};

export const useUser = (): AuthContextType => {

  const context = useContext(AuthContext);

  if (!context) {

    throw new Error('useUser must be used within a UserProvider');

  }

  return context;

};

Usercontext.tsr file code

import \* as React from "react"

const MOBILE\_BREAKPOINT = 768

export function useIsMobile() {

  const [isMobile, setIsMobile] = React.useState<boolean | undefined>(undefined)

  React.useEffect(() => {

    const mql = window.matchMedia(`(max-width: ${MOBILE\_BREAKPOINT - 1}px)`)

    const onChange = () => {

      setIsMobile(window.innerWidth < MOBILE\_BREAKPOINT)

    }

    mql.addEventListener("change", onChange)

    setIsMobile(window.innerWidth < MOBILE\_BREAKPOINT)

    return () => mql.removeEventListener("change", onChange)

  }, [])

  return !!isMobile

}

Use-moblie.tsx file code

import \* as React from "react"

import type {

  ToastActionElement,

  ToastProps,

} from "@/components/ui/toast"

const TOAST\_LIMIT = 1

const TOAST\_REMOVE\_DELAY = 1000000

type ToasterToast = ToastProps & {

  id: string

  title?: React.ReactNode

  description?: React.ReactNode

  action?: ToastActionElement

}

const actionTypes = {

  ADD\_TOAST: "ADD\_TOAST",

  UPDATE\_TOAST: "UPDATE\_TOAST",

  DISMISS\_TOAST: "DISMISS\_TOAST",

  REMOVE\_TOAST: "REMOVE\_TOAST",

} as const

let count = 0

function genId() {

  count = (count + 1) % Number.MAX\_SAFE\_INTEGER

  return count.toString()

}

type ActionType = typeof actionTypes

type Action =

  | {

      type: ActionType["ADD\_TOAST"]

      toast: ToasterToast

    }

  | {

      type: ActionType["UPDATE\_TOAST"]

      toast: Partial<ToasterToast>

    }

  | {

      type: ActionType["DISMISS\_TOAST"]

      toastId?: ToasterToast["id"]

    }

  | {

      type: ActionType["REMOVE\_TOAST"]

      toastId?: ToasterToast["id"]

    }

interface State {

  toasts: ToasterToast[]

}

const toastTimeouts = new Map<string, ReturnType<typeof setTimeout>>()

const addToRemoveQueue = (toastId: string) => {

  if (toastTimeouts.has(toastId)) {

    return

  }

  const timeout = setTimeout(() => {

    toastTimeouts.delete(toastId)

    dispatch({

      type: "REMOVE\_TOAST",

      toastId: toastId,

    })

  }, TOAST\_REMOVE\_DELAY)

  toastTimeouts.set(toastId, timeout)

}

export const reducer = (state: State, action: Action): State => {

  switch (action.type) {

    case "ADD\_TOAST":

      return {

        ...state,

        toasts: [action.toast, ...state.toasts].slice(0, TOAST\_LIMIT),

      }

    case "UPDATE\_TOAST":

      return {

        ...state,

        toasts: state.toasts.map((t) =>

          t.id === action.toast.id ? { ...t, ...action.toast } : t

        ),

      }

    case "DISMISS\_TOAST": {

      const { toastId } = action

      // ! Side effects ! - This could be extracted into a dismissToast() action,

      // but I'll keep it here for simplicity

      if (toastId) {

        addToRemoveQueue(toastId)

      } else {

        state.toasts.forEach((toast) => {

          addToRemoveQueue(toast.id)

        })

      }

      return {

        ...state,

        toasts: state.toasts.map((t) =>

          t.id === toastId || toastId === undefined

            ? {

                ...t,

                open: false,

              }

            : t

        ),

      }

    }

    case "REMOVE\_TOAST":

      if (action.toastId === undefined) {

        return {

          ...state,

          toasts: [],

        }

      }

      return {

        ...state,

        toasts: state.toasts.filter((t) => t.id !== action.toastId),

      }

  }

}

const listeners: Array<(state: State) => void> = []

let memoryState: State = { toasts: [] }

function dispatch(action: Action) {

  memoryState = reducer(memoryState, action)

  listeners.forEach((listener) => {

    listener(memoryState)

  })

}

type Toast = Omit<ToasterToast, "id">

function toast({ ...props }: Toast) {

  const id = genId()

  const update = (props: ToasterToast) =>

    dispatch({

      type: "UPDATE\_TOAST",

      toast: { ...props, id },

    })

  const dismiss = () => dispatch({ type: "DISMISS\_TOAST", toastId: id })

  dispatch({

    type: "ADD\_TOAST",

    toast: {

      ...props,

      id,

      open: true,

      onOpenChange: (open) => {

        if (!open) dismiss()

      },

    },

  })

  return {

    id: id,

    dismiss,

    update,

  }

}

function useToast() {

  const [state, setState] = React.useState<State>(memoryState)

  React.useEffect(() => {

    listeners.push(setState)

    return () => {

      const index = listeners.indexOf(setState)

      if (index > -1) {

        listeners.splice(index, 1)

      }

    }

  }, [state])

  return {

    ...state,

    toast,

    dismiss: (toastId?: string) => dispatch({ type: "DISMISS\_TOAST", toastId }),

  }

}

export { useToast, toast }

use-toast.ts file code

import { createContext, useContext, useState, useEffect, ReactNode } from 'react';

export type User = {

  name: string;

  email: string;

  avatarUrl?: string;

};

type AuthContextType = {

  user: User | null;

  loading: boolean;

  login: (userData: User) => void;

  logout: () => void;

};

const AuthContext = createContext<AuthContextType | undefined>(undefined);

export const AuthProvider = ({ children }: { children: ReactNode }) => {

  const [user, setUser] = useState<User | null>(null);

  const [loading, setLoading] = useState(true);

  useEffect(() => {

    const stored = localStorage.getItem('user');

    if (stored) {

      setUser(JSON.parse(stored));

    }

    setLoading(false);

  }, []);

  const login = (userData: User) => {

    localStorage.setItem('user', JSON.stringify(userData));

    setUser(userData);

  };

  const logout = () => {

    localStorage.removeItem('user');

    setUser(null);

  };

  return (

    <AuthContext.Provider value={{ user, loading, login, logout }}>

      {children}

    </AuthContext.Provider>

  );

};

export const useAuth = () => {

  const context = useContext(AuthContext);

  if (!context) {

    throw new Error('useAuth must be used within an AuthProvider');

  }

  return context;

};

useAuth.tsx file code

// This file is automatically generated. Do not edit it directly.

import { createClient } from '@supabase/supabase-js';

import type { Database } from './types';

const SUPABASE\_URL = "https://kiisanllgblmgswxfluo.supabase.co";

const SUPABASE\_PUBLISHABLE\_KEY = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImtpaXNhbmxsZ2JsbWdzd3hmbHVvIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NTIzMjAyMjUsImV4cCI6MjA2Nzg5NjIyNX0.D0mZQtBfRiWZFNXQc-0yMA4DgdVx1DKI6gyUybffRUY";

// Import the supabase client like this:

// import { supabase } from "@/integrations/supabase/client";

export const supabase = createClient<Database>(SUPABASE\_URL, SUPABASE\_PUBLISHABLE\_KEY, {

  auth: {

    storage: localStorage,

    persistSession: true,

    autoRefreshToken: true,

  }

});

Client.ts file code

export type Json =

  | string

  | number

  | boolean

  | null

  | { [key: string]: Json | undefined }

  | Json[]

export type Database = {

  // Allows to automatically instanciate createClient with right options

  // instead of createClient<Database, { PostgrestVersion: 'XX' }>(URL, KEY)

  \_\_InternalSupabase: {

    PostgrestVersion: "12.2.3 (519615d)"

  }

  public: {

    Tables: {

      [\_ in never]: never

    }

    Views: {

      [\_ in never]: never

    }

    Functions: {

      [\_ in never]: never

    }

    Enums: {

      [\_ in never]: never

    }

    CompositeTypes: {

      [\_ in never]: never

    }

  }

}

type DatabaseWithoutInternals = Omit<Database, "\_\_InternalSupabase">

type DefaultSchema = DatabaseWithoutInternals[Extract<keyof Database, "public">]

export type Tables<

  DefaultSchemaTableNameOrOptions extends

    | keyof (DefaultSchema["Tables"] & DefaultSchema["Views"])

    | { schema: keyof DatabaseWithoutInternals },

  TableName extends DefaultSchemaTableNameOrOptions extends {

    schema: keyof DatabaseWithoutInternals

  }

    ? keyof (DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Tables"] &

        DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Views"])

    : never = never,

> = DefaultSchemaTableNameOrOptions extends {

  schema: keyof DatabaseWithoutInternals

}

  ? (DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Tables"] &

      DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Views"])[TableName] extends {

      Row: infer R

    }

    ? R

    : never

  : DefaultSchemaTableNameOrOptions extends keyof (DefaultSchema["Tables"] &

        DefaultSchema["Views"])

    ? (DefaultSchema["Tables"] &

        DefaultSchema["Views"])[DefaultSchemaTableNameOrOptions] extends {

        Row: infer R

      }

      ? R

      : never

    : never

export type TablesInsert<

  DefaultSchemaTableNameOrOptions extends

    | keyof DefaultSchema["Tables"]

    | { schema: keyof DatabaseWithoutInternals },

  TableName extends DefaultSchemaTableNameOrOptions extends {

    schema: keyof DatabaseWithoutInternals

  }

    ? keyof DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Tables"]

    : never = never,

> = DefaultSchemaTableNameOrOptions extends {

  schema: keyof DatabaseWithoutInternals

}

  ? DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Tables"][TableName] extends {

      Insert: infer I

    }

    ? I

    : never

  : DefaultSchemaTableNameOrOptions extends keyof DefaultSchema["Tables"]

    ? DefaultSchema["Tables"][DefaultSchemaTableNameOrOptions] extends {

        Insert: infer I

      }

      ? I

      : never

    : never

export type TablesUpdate<

  DefaultSchemaTableNameOrOptions extends

    | keyof DefaultSchema["Tables"]

    | { schema: keyof DatabaseWithoutInternals },

  TableName extends DefaultSchemaTableNameOrOptions extends {

    schema: keyof DatabaseWithoutInternals

  }

    ? keyof DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Tables"]

    : never = never,

> = DefaultSchemaTableNameOrOptions extends {

  schema: keyof DatabaseWithoutInternals

}

  ? DatabaseWithoutInternals[DefaultSchemaTableNameOrOptions["schema"]]["Tables"][TableName] extends {

      Update: infer U

    }

    ? U

    : never

  : DefaultSchemaTableNameOrOptions extends keyof DefaultSchema["Tables"]

    ? DefaultSchema["Tables"][DefaultSchemaTableNameOrOptions] extends {

        Update: infer U

      }

      ? U

      : never

    : never

export type Enums<

  DefaultSchemaEnumNameOrOptions extends

    | keyof DefaultSchema["Enums"]

    | { schema: keyof DatabaseWithoutInternals },

  EnumName extends DefaultSchemaEnumNameOrOptions extends {

    schema: keyof DatabaseWithoutInternals

  }

    ? keyof DatabaseWithoutInternals[DefaultSchemaEnumNameOrOptions["schema"]]["Enums"]

    : never = never,

> = DefaultSchemaEnumNameOrOptions extends {

  schema: keyof DatabaseWithoutInternals

}

  ? DatabaseWithoutInternals[DefaultSchemaEnumNameOrOptions["schema"]]["Enums"][EnumName]

  : DefaultSchemaEnumNameOrOptions extends keyof DefaultSchema["Enums"]

    ? DefaultSchema["Enums"][DefaultSchemaEnumNameOrOptions]

    : never

export type CompositeTypes<

  PublicCompositeTypeNameOrOptions extends

    | keyof DefaultSchema["CompositeTypes"]

    | { schema: keyof DatabaseWithoutInternals },

  CompositeTypeName extends PublicCompositeTypeNameOrOptions extends {

    schema: keyof DatabaseWithoutInternals

  }

    ? keyof DatabaseWithoutInternals[PublicCompositeTypeNameOrOptions["schema"]]["CompositeTypes"]

    : never = never,

> = PublicCompositeTypeNameOrOptions extends {

  schema: keyof DatabaseWithoutInternals

}

  ? DatabaseWithoutInternals[PublicCompositeTypeNameOrOptions["schema"]]["CompositeTypes"][CompositeTypeName]

  : PublicCompositeTypeNameOrOptions extends keyof DefaultSchema["CompositeTypes"]

    ? DefaultSchema["CompositeTypes"][PublicCompositeTypeNameOrOptions]

    : never

export const Constants = {

  public: {

    Enums: {},

  },

} as const

Types.ts file code

import { clsx, type ClassValue } from "clsx"

import { twMerge } from "tailwind-merge"

export function cn(...inputs: ClassValue[]) {

  return twMerge(clsx(inputs))

}

Utils.ts file code

import React from 'react';

import styled from 'styled-components';

import { motion } from 'framer-motion';

import ResumeCard from '../components/ResumeCard';

const PageWrapper = styled(motion.div)`

  background-color: #0f172a;

  min-height: 100vh;

  padding: 2rem;

  color: white;

`;

const Header = styled.div`

  display: flex;

  justify-content: space-between;

  align-items: center;

  flex-wrap: wrap;

`;

const Greeting = styled.div`

  font-size: 1.8rem;

  font-weight: 600;

`;

const DateText = styled.p`

  font-size: 1rem;

  color: #94a3b8;

`;

const UploadButton = styled.button`

  background-color: #2563eb;

  color: white;

  border: none;

  padding: 0.75rem 1.25rem;

  border-radius: 8px;

  font-weight: 600;

  font-size: 1rem;

  cursor: pointer;

  transition: background 0.3s ease;

  &:hover {

    background-color: #1d4ed8;

  }

`;

const Grid = styled.div`

  display: grid;

  grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));

  gap: 1.5rem;

  margin-top: 2rem;

`;

const SectionTitle = styled.h2`

  margin-top: 3rem;

  font-size: 1.4rem;

  font-weight: 500;

`;

const Dashboard: React.FC = () => {

  const resumeHistory = [

    { id: 1, title: "Frontend Developer Resume", score: 82, date: "2025-07-20" },

    { id: 2, title: "React Engineer Resume", score: 76, date: "2025-07-18" },

    { id: 3, title: "Full Stack Resume", score: 90, date: "2025-07-15" },

  ];

  const today = new Date().toLocaleDateString();

  return (

    <PageWrapper

      initial={{ opacity: 0 }}

      animate={{ opacity: 1 }}

      transition={{ duration: 0.5 }}

    >

      <Header>

        <div>

          <Greeting>Welcome, Sanjay 👋</Greeting>

          <DateText>Today is {today}</DateText>

        </div>

        <UploadButton onClick={() => window.location.href = '/upload'}>

          Upload Resume

        </UploadButton>

      </Header>

      <SectionTitle>Recent Scores</SectionTitle>

      <Grid>

        {resumeHistory.map((r) => (

          <ResumeCard key={r.id} title={r.title} date={r.date} score={r.score} />

        ))}

      </Grid>

    </PageWrapper>

  );

};

export default Dashboard;

Dashboard.tsx file code

// Upload.tsx

export default function History() {

  return <div>Upload Page</div>;

}

History.tsx file code

import { Link } from "react-router-dom";

import Navbar from '../components/Navbar/Navbar';

import InfoSection from '../components/InfoSection/InfoSection';

export default function LandingPage() {

  return (

    <>

      <Navbar />

      {/\* 💡 Hero Section \*/}

      <section className="min-h-screen flex items-center justify-center bg-gradient-to-br from-white to-blue-50 px-4">

        <div className="max-w-2xl text-center py-12">

          <h1 className="text-5xl font-extrabold text-gray-900 mb-6 leading-tight">

            ATS Resume Checker

          </h1>

          <p className="text-lg text-gray-600 mb-8">

            Check how well your resume matches the job description using our AI-based scoring system.

          </p>

          <div className="flex justify-center gap-4">

            <Link to="/login">

              <button className="bg-blue-600 hover:bg-blue-700 text-white font-semibold px-6 py-3 rounded shadow transition">

                Login

              </button>

            </Link>

            <Link to="/register">

              <button className="border border-blue-600 text-blue-600 hover:bg-blue-50 font-semibold px-6 py-3 rounded transition">

                Register

              </button>

            </Link>

          </div>

        </div>

      </section>

      {/\* ✅ GitHub-style InfoSection \*/}

      <InfoSection

        primary

        lightBg={false}

        topLine="Top Features"

        lightTopLine

        headline="Instant Resume Scoring"

        lightText

        lightTextDesc

        description="Upload your resume and get instant feedback based on the job description using our smart AI system."

        buttonLabel="Get Started"

        img="/info1.png"

        alt="Resume Illustration"

        imgStart={false}

        start={false}

      />

    </>

  );

}

Landingpage.tsx file code

import styled, { keyframes } from "styled-components";

export const LoginContainer = styled.div`

  min-height: 100vh;

  display: flex;

  align-items: center;

  justify-content: center;

  background: radial-gradient(ellipse at top left, #c2dfff, #e0d3ff, #fce2ff);

  overflow: hidden;

  font-family: 'DM Sans', 'Inter', sans-serif;

  position: relative;

`;

export const LoginCard = styled.div`

  backdrop-filter: blur(20px);

  background: rgba(255, 255, 255, 0.15);

  border: 1px solid rgba(255, 255, 255, 0.25);

  box-shadow: 0 8px 40px rgba(0, 0, 0, 0.15);

  padding: 3rem 2rem;

  border-radius: 16px;

  max-width: 400px;

  width: 100%;

  color: #111;

`;

export const Heading = styled.h2`

  text-align: center;

  font-size: 2rem;

  font-weight: 700;

  color: #1a1a1a;

  margin-bottom: 2rem;

`;

export const AnimatedForm = styled.form`

  display: flex;

  flex-direction: column;

  gap: 1.5rem;

`;

export const InputWrapper = styled.div`

  position: relative;

  width: 100%;

`;

export const StyledInput = styled.input`

  width: 100%;

  padding: 1rem 1rem 1rem 3rem;

  border: 1px solid #ccc;

  background: rgba(255, 255, 255, 0.5);

  border-radius: 10px;

  outline: none;

  font-size: 1rem;

  transition: border-color 0.3s, background 0.3s;

  &:focus {

    border-color: #7f56d9;

    background: rgba(255, 255, 255, 0.9);

  }

  &:focus ~ label,

  &:not(:placeholder-shown) ~ label {

    top: -10px;

    left: 2.5rem;

    font-size: 0.75rem;

    color: #7f56d9;

  }

`;

export const StyledLabel = styled.label`

  position: absolute;

  top: 50%;

  left: 3rem;

  transform: translateY(-50%);

  color: #777;

  font-size: 1rem;

  transition: 0.2s ease;

  pointer-events: none;

`;

export const IconWrapper = styled.div`

  position: absolute;

  top: 50%;

  left: 0.9rem;

  transform: translateY(-50%);

  font-size: 1.2rem;

  color: #999;

`;

export const SubmitButton = styled.button`

  background: linear-gradient(to right, #7f56d9, #5e60ce);

  color: white;

  border: none;

  border-radius: 8px;

  padding: 0.9rem;

  font-size: 1rem;

  font-weight: 600;

  cursor: pointer;

  transition: all 0.3s ease;

  &:hover {

    box-shadow: 0 0 10px rgba(94, 96, 206, 0.5);

    transform: translateY(-1px);

  }

  &:disabled {

    opacity: 0.7;

    cursor: not-allowed;

  }

`;

export const Divider = styled.div`

  text-align: center;

  margin: 1.5rem 0;

  font-size: 0.85rem;

  color: #666;

  position: relative;

  &:before,

  &:after {

    content: "";

    height: 1px;

    width: 40%;

    background: #ccc;

    position: absolute;

    top: 50%;

  }

  &:before {

    left: 0;

  }

  &:after {

    right: 0;

  }

`;

export const OAuthWrapper = styled.div`

  display: flex;

  justify-content: center;

`;

export const Spinner = styled.div`

  border: 3px solid rgba(255, 255, 255, 0.3);

  border-top: 3px solid #fff;

  border-radius: 50%;

  width: 18px;

  height: 18px;

  animation: spin 0.6s linear infinite;

  @keyframes spin {

    to {

      transform: rotate(360deg);

    }

  }

`;

export const DarkToggle = styled.div``; // optional: placeholder for toggle logic

login.elements.tsx file code

import { useState } from "react";

import { useNavigate } from "react-router-dom";

import { loginUser } from "@/services/api";

import { useAuth } from "../hooks/useAuth";

import { GoogleLogin } from "@react-oauth/google";

import {

  LoginContainer,

  LoginCard,

  DarkToggle,

  AnimatedForm,

  InputWrapper,

  StyledInput,

  StyledLabel,

  SubmitButton,

  Heading,

  Divider,

  OAuthWrapper,

  IconWrapper,

  Spinner

} from "./Login.elements";

import { motion } from "framer-motion";

import { HiOutlineEnvelope, HiOutlineLockClosed } from "react-icons/hi2";

export default function Login() {

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const [loading, setLoading] = useState(false);

  const navigate = useNavigate();

  const { login } = useAuth();

  const handleLogin = async (e: React.FormEvent) => {

    e.preventDefault();

    try {

      setLoading(true);

      const token = await loginUser({ email, password });

      const userData = { email };

      localStorage.setItem("token", token);

      login(userData);

      navigate("/dashboard");

    } catch (err) {

      alert("Login failed. Please check credentials.");

      console.error(err);

    } finally {

      setLoading(false);

    }

  };

  return (

    <LoginContainer>

      <motion.div

        initial={{ opacity: 0, y: 40 }}

        animate={{ opacity: 1, y: 0 }}

        transition={{ duration: 0.6 }}

      >

        <DarkToggle />

        <LoginCard>

          <Heading>Login to Your Account</Heading>

          <AnimatedForm onSubmit={handleLogin}>

            <InputWrapper>

              <IconWrapper><HiOutlineEnvelope /></IconWrapper>

              <StyledInput

                type="email"

                value={email}

                onChange={(e) => setEmail(e.target.value)}

                required

              />

              <StyledLabel>Email</StyledLabel>

            </InputWrapper>

            <InputWrapper>

              <IconWrapper><HiOutlineLockClosed /></IconWrapper>

              <StyledInput

                type="password"

                value={password}

                onChange={(e) => setPassword(e.target.value)}

                required

              />

              <StyledLabel>Password</StyledLabel>

            </InputWrapper>

            <SubmitButton type="submit" disabled={loading}>

              {loading ? <Spinner /> : "Login"}

            </SubmitButton>

          </AnimatedForm>

          <Divider>or</Divider>

          <OAuthWrapper>

            <GoogleLogin

              onSuccess={(credentialResponse) => {

                const token = credentialResponse.credential;

                if (token) {

                  localStorage.setItem("google\_token", token);

                  navigate("/dashboard");

                }

              }}

              onError={() => {

                alert("Google login failed");

              }}

            />

          </OAuthWrapper>

        </LoginCard>

      </motion.div>

    </LoginContainer>

  );

}

Login.tsx file code

export function Register() {

  return (

    <div>

      <h2>Register Page</h2>

      {/\* form here \*/}

    </div>

  );

}

export default Register;

register.tsx file code

import { useEffect, useState } from 'react';

export default function Score() {

  const [score, setScore] = useState<number | null>(null);

  useEffect(() => {

    const storedScore = localStorage.getItem('score');

    if (storedScore) {

      setScore(Number(storedScore));

    }

  }, []);

  return (

    <div style={{ textAlign: 'center', marginTop: '50px' }}>

      {score !== null ? (

        <h1>Your ATS Score: {score}%</h1>

      ) : (

        <p>Loading score...</p>

      )}

    </div>

  );

}

Score.tsx file code

import { useState } from 'react';

import { useNavigate } from 'react-router-dom';

import axios from 'axios';

import { CloudArrowUpIcon } from '@heroicons/react/24/solid';

import { motion } from 'framer-motion';

import Navbar from '../components/Navbar/Navbar'; // ✅ Adjust the path if needed

const Upload = () => {

  const [resume, setResume] = useState<File | null>(null);

  const [jobDesc, setJobDesc] = useState<File | null>(null);

  const [loading, setLoading] = useState(false);

  const navigate = useNavigate();

  const handleSubmit = async (e: React.FormEvent) => {

    e.preventDefault();

    if (!resume || !jobDesc) {

      alert('Please upload both files.');

      return;

    }

    const formData = new FormData();

    formData.append('resume', resume);

    formData.append('job', jobDesc);

    try {

      setLoading(true);

      const token = localStorage.getItem('token');

      const response = await axios.post('http://localhost:5000/api/score', formData, {

        headers: {

          'Authorization': `Bearer ${token}`,

        },

      });

      const score = response.data.score;

      localStorage.setItem('score', score.toString());

      navigate('/score');

    } catch (err) {

      alert('Error uploading files. Please try again.');

      console.error(err);

    } finally {

      setLoading(false);

    }

  };

  return (

    <>

      <Navbar />

      <div className="min-h-screen bg-gradient-to-br from-indigo-200 via-purple-300 to-pink-200 flex items-center justify-center px-4">

        <motion.div

          initial={{ opacity: 0, y: 60 }}

          animate={{ opacity: 1, y: 0 }}

          transition={{ duration: 0.6 }}

          className="backdrop-blur-md bg-white/30 border border-white/40 rounded-xl shadow-xl p-8 w-full max-w-xl"

        >

          <div className="text-center mb-6">

            <CloudArrowUpIcon className="h-12 w-12 text-indigo-600 mx-auto" />

            <h2 className="text-3xl font-bold text-gray-800 mt-2">Upload Your Resume</h2>

            <p className="text-gray-600 mt-1">Match your resume to the job description</p>

          </div>

          <form onSubmit={handleSubmit} className="space-y-4">

            {/\* Resume File Input \*/}

            <label className="flex items-center justify-between gap-3 px-4 py-3 bg-white/50 hover:bg-white/70 transition border border-gray-300 rounded-lg cursor-pointer">

              <div className="text-left flex-1">

                <span className="block text-gray-700 font-medium">Resume</span>

                <span className="text-sm text-gray-600">

                  {resume ? resume.name : "No file selected"}

                </span>

              </div>

              <input

                type="file"

                accept=".pdf,.doc,.docx"

                onChange={(e) => setResume(e.target.files?.[0] || null)}

                className="hidden"

              />

              <CloudArrowUpIcon className="h-5 w-5 text-indigo-500" />

            </label>

            {/\* Job Description File Input \*/}

            <label className="flex items-center justify-between gap-3 px-4 py-3 bg-white/50 hover:bg-white/70 transition border border-gray-300 rounded-lg cursor-pointer">

              <div className="text-left flex-1">

                <span className="block text-gray-700 font-medium">Job Description</span>

                <span className="text-sm text-gray-600">

                  {jobDesc ? jobDesc.name : "No file selected"}

                </span>

              </div>

              <input

                type="file"

                accept=".txt,.pdf"

                onChange={(e) => setJobDesc(e.target.files?.[0] || null)}

                className="hidden"

              />

              <CloudArrowUpIcon className="h-5 w-5 text-indigo-500" />

            </label>

            <button

              type="submit"

              disabled={loading}

              className="w-full flex items-center justify-center gap-2 bg-gradient-to-r from-indigo-500 to-purple-500 text-white font-semibold py-3 px-6 rounded-lg shadow-md hover:from-indigo-600 hover:to-purple-600 transition"

            >

              {loading ? (

                <svg className="animate-spin h-5 w-5 text-white" viewBox="0 0 24 24">

                  <circle className="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" strokeWidth="4" />

                  <path className="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8v8z" />

                </svg>

              ) : (

                'Submit'

              )}

            </button>

          </form>

        </motion.div>

      </div>

    </>

  );

};

export default Upload;

upload.tsx file code

export async function loginUser(data: { email: string; password: string }) {

  const response = await fetch("http://localhost:5000/api/auth/login", {

    method: "POST",

    headers: {

      "Content-Type": "application/json",

    },

    body: JSON.stringify(data),

  });

  if (!response.ok) {

    throw new Error("Login failed");

  }

  const result = await response.json();

  return result.token;

}

Api.ts file code  
  
export interface Resume {

  id: number;

  title: string;

  uploadDate: string;

  score: number;

}

Resume.d.ts file code

#root {

  max-width: 1280px;

  margin: 0 auto;

  padding: 2rem;

  text-align: center;

}

.logo {

  height: 6em;

  padding: 1.5em;

  will-change: filter;

  transition: filter 300ms;

}

.logo:hover {

  filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

  filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

@media (prefers-reduced-motion: no-preference) {

  a:nth-of-type(2) .logo {

    animation: logo-spin infinite 20s linear;

  }

}

.card {

  padding: 2em;

}

.read-the-docs {

  color: #888;

}

App.css file code

// src/App.tsx

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

import LandingPage from './pages/LandingPage';

import Login from './pages/Login';

import Register from './pages/Register';

import Upload from './pages/Upload';

import Score from './pages/Score';

import History from './pages/History';

import ProtectedRoute from './components/ProtectedRoute';

import Dashboard from "./pages/Dashboard";

function App() {

  return (

    <Router>

      <Routes>

        <Route path="/" element={<LandingPage />} />

        <Route path="/dashboard" element={<Dashboard />} />

        <Route path="/upload" element={<Upload />} />

        <Route path="/login" element={<Login />} />

        <Route path="/register" element={<Register />} />

        {/\* Protected routes \*/}

        <Route

          path="/score"

          element={

            <ProtectedRoute>

              <Score />

            </ProtectedRoute>

          }

        />

        <Route

          path="/history"

          element={

            <ProtectedRoute>

              <History />

            </ProtectedRoute>

          }

        />

      </Routes>

    </Router>

  );

}

export default App;

app.tsx file code  
  
@tailwind base;

@tailwind components;

@tailwind utilities;

/\* Definition of the design system. All colors, gradients, fonts, etc should be defined here.

All colors MUST be HSL.

\*/

@layer base {

  :root {

    --background: 0 0% 100%;

    --foreground: 222.2 84% 4.9%;

    --card: 0 0% 100%;

    --card-foreground: 222.2 84% 4.9%;

    --popover: 0 0% 100%;

    --popover-foreground: 222.2 84% 4.9%;

    --primary: 222.2 47.4% 11.2%;

    --primary-foreground: 210 40% 98%;

    --secondary: 210 40% 96.1%;

    --secondary-foreground: 222.2 47.4% 11.2%;

    --muted: 210 40% 96.1%;

    --muted-foreground: 215.4 16.3% 46.9%;

    --accent: 210 40% 96.1%;

    --accent-foreground: 222.2 47.4% 11.2%;

    --destructive: 0 84.2% 60.2%;

    --destructive-foreground: 210 40% 98%;

    --border: 214.3 31.8% 91.4%;

    --input: 214.3 31.8% 91.4%;

    --ring: 222.2 84% 4.9%;

    --radius: 0.5rem;

    --sidebar-background: 0 0% 98%;

    --sidebar-foreground: 240 5.3% 26.1%;

    --sidebar-primary: 240 5.9% 10%;

    --sidebar-primary-foreground: 0 0% 98%;

    --sidebar-accent: 240 4.8% 95.9%;

    --sidebar-accent-foreground: 240 5.9% 10%;

    --sidebar-border: 220 13% 91%;

    --sidebar-ring: 217.2 91.2% 59.8%;

  }

  .dark {

    --background: 222.2 84% 4.9%;

    --foreground: 210 40% 98%;

    --card: 222.2 84% 4.9%;

    --card-foreground: 210 40% 98%;

    --popover: 222.2 84% 4.9%;

    --popover-foreground: 210 40% 98%;

    --primary: 210 40% 98%;

    --primary-foreground: 222.2 47.4% 11.2%;

    --secondary: 217.2 32.6% 17.5%;

    --secondary-foreground: 210 40% 98%;

    --muted: 217.2 32.6% 17.5%;

    --muted-foreground: 215 20.2% 65.1%;

    --accent: 217.2 32.6% 17.5%;

    --accent-foreground: 210 40% 98%;

    --destructive: 0 62.8% 30.6%;

    --destructive-foreground: 210 40% 98%;

    --border: 217.2 32.6% 17.5%;

    --input: 217.2 32.6% 17.5%;

    --ring: 212.7 26.8% 83.9%;

    --sidebar-background: 240 5.9% 10%;

    --sidebar-foreground: 240 4.8% 95.9%;

    --sidebar-primary: 224.3 76.3% 48%;

    --sidebar-primary-foreground: 0 0% 100%;

    --sidebar-accent: 240 3.7% 15.9%;

    --sidebar-accent-foreground: 240 4.8% 95.9%;

    --sidebar-border: 240 3.7% 15.9%;

    --sidebar-ring: 217.2 91.2% 59.8%;

  }

}

Index.css file code

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

import './index.css';

import { GoogleOAuthProvider } from '@react-oauth/google';

import { AuthProvider } from './hooks/useAuth'; // ✅ Adjust path if needed

ReactDOM.createRoot(document.getElementById('root')!).render(

  <React.StrictMode>

    <GoogleOAuthProvider clientId="567550346745-ktd6isvo0qurnltkcubnco6j30f7lt3e.apps.googleusercontent.com">

      <AuthProvider>

        <App />

      </AuthProvider>

    </GoogleOAuthProvider>

  </React.StrictMode>

);

Main.tsx file code

/// <reference types="vite/client" />

Vite-env.d.ts file code