**File Structure for SmartDentAi Web application:**

SmartDentAi-Project/

Appointment/

node\_modules/

public/

vite.svg

src/

assets/

components/

Appointment.jsx

Banner.jsx

Footer.jsx

Header.jsx

MyAppointments.jsx

Navbar.jsx

RelatedDoctors.jsx

SpecialityMenu.jsx

Topdoctors.jsx

context/

AppContext.jsx

pages/

About.jsx

Appointment.jsx

Contact.jsx

Doctors.jsx

Home.jsx

Login.jsx

MyAppointments.jsx

MyProfile.jsx

App.jsx

Index.css

Main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock-json

package-json

postcss.config.js

README.md

tailwind.config.js

vercel.json

Vite.config.js

Chatbot/

myenv/

node\_modules/

.env

.gitignore

index.html

loader.gif

package-lock.json

package.json

README.md

server.js

model/

Dental\_model.h5

dental\_radiography\_best\_model.pth

oral\_radiography\_best\_model.pth

Xray\_Pred\_model.h5

plot\_metrics.py

train\_dental\_radiography.py

train\_oral\_disease.py

code.ipynb

myenv/

Include/

Lib/

Site-packages/

Scripts/

Pyvenv.cfg

Oral/

model/

oral\_radiography\_best\_model.pth

oral\_radiography\_best\_model.h5

oral\_model.pth

static/

(images of oral)

template/

index.html

test\_image/

(testing images)

venv/

Include/

Lib/

Site-packages/

Scripts/

share/

Pyvenv.cfg

app.py

predict\_oral.ipynb

ortho/

code/

\_\_pycache\_\_/

checkpoints/

model\_epoch\_195.pth

content/

outputs/

(trained images)

static/

assets/

(images, icons)

outputs/

(output images)

uploads/

(uploaded images)

style.css

template/

index.html

uploads/

(upload image)

app.py

image\_alignment\_results.db

model.py

train.py

data/

test/

input/

(input images)

output/

(output images)

train/

input/

(trained input images)

target/

(trained targetes images)

Remote/

instance/

Video\_meeting.db

static/

styles.css

templates/

base.html

dashboard.html

join.html

login.html

meeting.html

register.html

.gitignore

LICENSE

main.py

README.md

static/

assets/

uploads/

(upload images)

(images, icons)

about\_us.css

chatbot.css

contact\_us.css

dentalimg.css

index.js

login.css

login.js

main.css

orthodontic.css

predict.css

profile.css

profile.js

register.js

scripts.js

Signup.css

styles.css

templates/

about\_us.html

contact.html

dentalimg.html

dentalxray.html

home.html

index.html

login.html

Main.html

oraldetection.html

orthodontic.html

predict.html

Profile.html

Signup.html

user.html

.env

app.py

oral\_disease\_confusion\_matrix.png

oral\_disease\_learning\_curves.png

oral\_disease\_metrics.csv

requirements.txt

server.py

test\_api.py

test\_your\_file.py

training.metrics.csv

**Full Web Application project code for SmartDentAi in HTML, CSS, JS and Python:**

**Appointment/src/components:**

**Appointment.jsx:**

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

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

import { AppContext } from '../context/AppContext';

const Appointment = () => {

    const { doctorId } = useParams();

    const navigate = useNavigate();

    const { doctors } = React.useContext(AppContext);

    const [doctor, setDoctor] = useState(null);

    const [selectedDate, setSelectedDate] = useState('');

    const [selectedTime, setSelectedTime] = useState('');

    const [bookingConfirmation, setBookingConfirmation] = useState(null);

    const [availableDates, setAvailableDates] = useState([]);

    const [timeSlots, setTimeSlots] = useState([]);

    // Generate dynamic dates and time slots

    useEffect(() => {

        const generateDatesAndSlots = () => {

            const dates = [];

            const times = [];

            const today = new Date();

            const daysOfWeek = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];

            // Generate dates for the next 7 days

            for (let i = 0; i < 7; i++) {

                const date = new Date(today);

                date.setDate(today.getDate() + i);

                const dayName = daysOfWeek[date.getDay()];

                const day = date.getDate();

                const month = date.getMonth() + 1; // Months are 0-based in JS

                const year = date.getFullYear();

                const formattedDate = `${dayName} ${day}\_${month}\_${year}`;

                dates.push(formattedDate);

            }

            // Generate time slots from 10:00 AM to 4:30 PM with 30-minute intervals

            const startHour = 10; // 10:00 AM

            const endHour = 16; // 4:30 PM

            for (let hour = startHour; hour <= endHour; hour++) {

                for (let minute = 0; minute < 60; minute += 30) {

                    if (hour === endHour && minute > 30) break; // Stop at 4:30 PM

                    const period = hour >= 12 ? 'PM' : 'AM';

                    const displayHour = hour > 12 ? hour - 12 : hour === 0 ? 12 : hour;

                    const formattedTime = `${displayHour}:${minute === 0 ? '00' : minute} ${period}`;

                    times.push(formattedTime);

                }

            }

            setAvailableDates(dates);

            setTimeSlots(times);

        };

        generateDatesAndSlots();

    }, []);

    // Test the proxy on component mount

    useEffect(() => {

        console.log('Appointment component mounted');

        const testApi = async () => {

            try {

                console.log('Testing proxy with /api/test...');

                const response = await fetch('/api/test', {

                    method: 'GET',

                    headers: {

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

                    },

                });

                if (!response.ok) {

                    throw new Error(`HTTP error! Status: ${response.status}`);

                }

                const data = await response.json();

                console.log('Test API response:', data);

            } catch (error) {

                console.error('Test API error:', error.message);

                console.error('Error details:', error);

            }

        };

        testApi();

    }, []);

    useEffect(() => {

        console.log('Fetching doctor data for ID:', doctorId);

        const doc = doctors.find(d => d.\_id === doctorId);

        if (doc) {

            console.log('Doctor found in context:', doc);

            setDoctor(doc);

        } else {

            console.log('Doctor not in context, fetching from backend...');

            fetch(`/api/doctors/${doctorId}`)

                .then(res => {

                    console.log('Doctor fetch response status:', res.status);

                    return res.json();

                })

                .then(data => {

                    console.log('Doctor fetch response data:', data);

                    setDoctor(data);

                })

                .catch(err => console.error('Error fetching doctor:', err));

        }

    }, [doctorId, doctors]);

    const handleBookAppointment = async () => {

        console.log('handleBookAppointment triggered');

        console.log('Selected Date:', selectedDate);

        console.log('Selected Time:', selectedTime);

        console.log('Doctor ID:', doctorId);

        if (!selectedDate || !selectedTime) {

            console.log('Validation failed: Date or time not selected');

            alert('Please select a date and time slot.');

            return;

        }

        try {

            console.log('Preparing to send booking request to:', `/api/book\_appointment/${doctorId}`);

            const response = await fetch(`/api/book\_appointment/${doctorId}`, {

                method: 'POST',

                headers: {

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

                },

                body: JSON.stringify({

                    date: selectedDate,

                    time: selectedTime,

                }),

            });

            console.log('Booking request sent, response status:', response.status);

            const result = await response.json();

            console.log('Booking response data:', result);

            if (response.ok) {

                console.log('Setting booking confirmation:', result);

                setBookingConfirmation(result);

            } else {

                console.log('Booking failed:', result.error);

                alert(`Error: ${result.error}`);

            }

        } catch (error) {

            console.error('Fetch Error in handleBookAppointment:', error.message);

            console.error('Error details:', error);

            alert(`Error: ${error.message}`);

        }

    };

    const handleBookAnother = () => {

        console.log('Booking another appointment');

        setSelectedDate('');

        setSelectedTime('');

        setBookingConfirmation(null);

    };

    const handleButtonClick = () => {

        console.log('Book appointment button clicked');

        handleBookAppointment();

    };

    if (!doctor) {

        console.log('Doctor not loaded yet, rendering loading state');

        return <div>Loading...</div>;

    }

    console.log('Rendering Appointment component with doctor:', doctor);

    console.log('bookingConfirmation state:', bookingConfirmation);

    return (

        <div className="p-4">

            {/\* Doctor Details \*/}

            <div className="flex border rounded-lg p-4">

                <img src={doctor.image} alt={doctor.name} className="w-32 h-32 rounded-lg mr-4" />

                <div>

                    <h2 className="text-xl font-semibold">{doctor.name}</h2>

                    <p className="text-gray-600">{doctor.speciality}</p>

                    <p className="mt-2">About: Dr. {doctor.name.split(' ')[1]} has a strong commitment...</p>

                    <p className="mt-2">Appointment fee: ${doctor.fee || 30}</p>

                </div>

            </div>

            {/\* Booking Section or Confirmation \*/}

            {bookingConfirmation ? (

                <div className="mt-4 p-4 bg-green-100 border border-green-300 rounded-lg">

                    <h3 className="text-lg font-medium text-green-700">{bookingConfirmation.message}</h3>

                    <p className="mt-2 text-gray-700">

                        You have booked an appointment with <strong>{bookingConfirmation.doctor\_name}</strong> on{' '}

                        <strong>{bookingConfirmation.date}</strong> at <strong>{bookingConfirmation.time}</strong>.

                    </p>

                    <div className="mt-4 flex gap-2">

                        <button

                            onClick={handleBookAnother}

                            className="bg-blue-500 text-white px-4 py-2 rounded-full"

                        >

                            Book Another Appointment

                        </button>

                        <button

                            onClick={() => navigate('/my-appointments')}

                            className="bg-gray-500 text-white px-4 py-2 rounded-full"

                        >

                            View My Appointments

                        </button>

                    </div>

                </div>

            ) : (

                <>

                    <div className="mt-4">

                        <h3 className="text-lg font-medium">Booking slots</h3>

                        <div className="flex gap-2 mt-2">

                            {availableDates.map(date => (

                                <button

                                    key={date}

                                    onClick={() => {

                                        setSelectedDate(date);

                                        setSelectedTime(''); // Reset time when date changes

                                        console.log('Selected date:', date);

                                    }}

                                    className={`px-4 py-2 rounded-full border ${

                                        selectedDate === date ? 'bg-blue-500 text-white' : 'bg-gray-100'

                                    }`}

                                >

                                    {date}

                                </button>

                            ))}

                        </div>

                        {selectedDate && (

                            <div className="flex gap-2 mt-2">

                                {timeSlots.map(time => (

                                    <button

                                        key={time}

                                        onClick={() => {

                                            setSelectedTime(time);

                                            console.log('Selected time:', time);

                                        }}

                                        className={`px-4 py-2 rounded-full border ${

                                            selectedTime === time ? 'bg-blue-500 text-white' : 'bg-gray-100'

                                        }`}

                                    >

                                        {time}

                                    </button>

                                ))}

                            </div>

                        )}

                    </div>

                    {console.log('Rendering booking section')}

                    <button

                        onClick={handleButtonClick}

                        className="mt-4 bg-blue-500 text-white px-6 py-2 rounded-full"

                    >

                        Book appointment

                    </button>

                </>

            )}

        </div>

    );

};

export default Appointment;

**Banner.jsx:**

import React from 'react'

import { assets } from '../assets/assets'

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

const Banner = () => {

    const navigate = useNavigate()

    return (

        <div className='flex bg-primary rounded-lg  px-6 sm:px-10 md:px-14 lg:px-12 my-20 md:mx-10'>

            {/\* ------- Left Side ------- \*/}

            <div className='flex-1 py-8 sm:py-10 md:py-16 lg:py-24 lg:pl-5'>

                <div className='text-xl sm:text-2xl md:text-3xl lg:text-5xl font-semibold text-white'>

                    <p>Book Appointment</p>

                    <p className='mt-4'>With 100+ Trusted Doctors</p>

                </div>

                <button onClick={() => { navigate('/login'); scrollTo(0, 0) }} className='bg-white text-sm sm:text-base text-gray-600 px-8 py-3 rounded-full mt-6 hover:scale-105 transition-all '>Create account</button>

            </div>

            {/\* ------- Right Side ------- \*/}

            <div className='hidden md:block md:w-1/2 lg:w-[370px] relative'>

                <img className='w-full absolute bottom-0 right-0 max-w-md' src={assets.appointment\_img} alt="" />

            </div>

        </div>

    )

}

export default Banner

**Footer.jsx:**

import React from 'react'

import { assets } from '../assets/assets'

const Footer = () => {

  return (

    <div className='md:mx-10'>

      <div className='flex flex-col sm:grid grid-cols-[3fr\_1fr\_1fr] gap-14 my-10  mt-40 text-sm'>

      </div>

    </div>

  )

}

export default Footer

**Header.jsx:**

import React from 'react'

import { assets } from '../assets/assets'

const Header = () => {

    return (

        <div className='flex flex-col md:flex-row flex-wrap bg-primary rounded-lg px-6 md:px-10 lg:px-20 '>

            {/\* --------- Header Left --------- \*/}

            <div className='md:w-1/2 flex flex-col items-start justify-center gap-4 py-10 m-auto md:py-[10vw] md:mb-[-30px]'>

                <p className='text-3xl md:text-4xl lg:text-5xl text-white font-semibold leading-tight md:leading-tight lg:leading-tight'>

                    Book Appointment <br />  With Trusted Doctors

                </p>

                <div className='flex flex-col md:flex-row items-center gap-3 text-white text-sm font-light'>

                    <img className='w-28' src={assets.group\_profiles} alt="" />

                    <p>Simply browse through our extensive list of trusted doctors, <br className='hidden sm:block' /> schedule your appointment hassle-free.</p>

                </div>

                <a href='#speciality' className='flex items-center gap-2 bg-white px-8 py-3 rounded-full text-gray-600 text-sm m-auto md:m-0 hover:scale-105 transition-all duration-300'>

                    Book appointment <img className='w-3' src={assets.arrow\_icon} alt="" />

                </a>

            </div>

            {/\* --------- Header Right --------- \*/}

            <div className='md:w-1/2 relative'>

                <img className='w-full md:absolute bottom-0 h-auto rounded-lg' src={assets.header\_img} alt="" />

            </div>

        </div>

    )

}

export default Header

**MyAppointments.jsx:**

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

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

const MyAppointments = () => {

    const navigate = useNavigate();

    const [appointments, setAppointments] = useState([]);

    const [error, setError] = useState(null);

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

    const fetchAppointments = async () => {

        setLoading(true);

        setError(null);

        try {

            const userId = "user1"; // Static for now; use auth in production

            console.log('Fetching appointments for user:', userId);

            const response = await fetch(`/api/my\_appointments/${userId}`, {

                method: 'GET',

                headers: {

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

                },

            });

            console.log('Fetch response status:', response.status);

            if (!response.ok) {

                throw new Error(`HTTP error! Status: ${response.status}`);

            }

            const data = await response.json();

            console.log('Fetched appointments:', data);

            if (data.error) {

                setError(data.error);

            } else {

                setAppointments(data);

            }

        } catch (err) {

            console.error('Fetch error:', err.message);

            console.error('Error details:', err);

            setError(err.message);

        } finally {

            setLoading(false);

        }

    };

    useEffect(() => {

        fetchAppointments();

    }, []);

    return (

        <div className="p-4">

            <h2 className="text-2xl font-semibold mb-4">My Appointments</h2>

            {loading ? (

                <p>Loading appointments...</p>

            ) : error ? (

                <div className="p-4 text-red-500">Error: {error}</div>

            ) : appointments.length === 0 ? (

                <p>No appointments booked yet.</p>

            ) : (

                <div className="grid gap-4">

                    {appointments.map((appt, index) => (

                        <div key={index} className="border rounded-lg p-4">

                            <p><strong>Doctor:</strong> {appt.doctor\_name}</p>

                            <p><strong>Date:</strong> {appt.date}</p>

                            <p><strong>Time:</strong> {appt.time}</p>

                        </div>

                    ))}

                </div>

            )}

            <div className="mt-4 flex gap-2">

                <button

                    onClick={fetchAppointments}

                    className="bg-blue-500 text-white px-4 py-2 rounded-full"

                >

                    Refresh Appointments

                </button>

                <button

                    onClick={() => navigate('/')}

                    className="bg-gray-500 text-white px-4 py-2 rounded-full"

                >

                    Back to Doctors

                </button>

            </div>

        </div>

    );

};

export default MyAppointments;

**Navbar.jsx:**

import React, { useState } from 'react'

import { assets } from '../assets/assets'

import { NavLink, useNavigate } from 'react-router-dom'

const Navbar = () => {

  const navigate = useNavigate()

  const [showMenu, setShowMenu] = useState(false)

  const [token, setToken] = useState(true)

  return (

    <div className='flex items-center justify-between text-sm py-4 mb-5 border-b border-b-gray-400'>

      <img onClick={() => navigate('/')} className='w-44 cursor-pointer' src={assets.logo} alt="" />

      <ul className='md:flex items-start gap-5 font-medium hidden'>

        <NavLink to='/doctors' >

          <li className='py-1'>ALL DOCTORS</li>

          <hr className='border-none outline-none h-0.5 bg-primary w-3/5 m-auto hidden' />

        </NavLink>

      </ul>

      <div className='flex items-center gap-4 '>

        {

          token

            ? <div className='flex items-center gap-2 cursor-pointer group relative'>

              <img className='w-8 rounded-full' src={assets.profile\_pic} alt="" />

              <img className='w-2.5' src={assets.dropdown\_icon} alt="" />

              <div className='absolute top-0 right-0 pt-14 text-base font-medium text-gray-600 z-20 hidden group-hover:block'>

                <div className='min-w-48 bg-stone-100 rounded flex flex-col gap-4 p-4'>

                  <p onClick={() => navigate('/my-profile')} className='hover:text-black cursor-pointer'>My Profile</p>

                  <p onClick={() => navigate('/my-appointments')} className='hover:text-black cursor-pointer'>My Appointments</p>

                  <p onClick={() => { setToken(false); navigate('/') }} className='hover:text-black cursor-pointer'>Logout</p>

                </div>

              </div>

            </div>

            : <button onClick={() => navigate('/login')} className='bg-primary text-white px-8 py-3 rounded-full font-light hidden md:block'>Create account</button>

        }

        <img onClick={() => setShowMenu(true)} className='w-6 md:hidden' src={assets.menu\_icon} alt="" />

        {/\* ---- Mobile Menu ---- \*/}

        <div className={`md:hidden right-0 top-0 bottom-0 z-20 overflow-hidden bg-white transition-all ${showMenu ? 'fixed w-full' : 'h-0 w-0'}`}>

          <div className='flex items-center justify-between px-5 py-6'>

            <img src={assets.logo} className='w-36' alt="" />

            <img onClick={() => setShowMenu(false)} src={assets.cross\_icon} className='w-7' alt="" />

          </div>

          <ul className='flex flex-col items-center gap-2 mt-5 px-5 text-lg font-medium'>

            <NavLink onClick={() => setShowMenu(false)} to='/'><p className='px-4 py-2 rounded full inline-block'>HOME</p></NavLink>

            <NavLink onClick={() => setShowMenu(false)} to='/doctors' ><p className='px-4 py-2 rounded full inline-block'>ALL DOCTORS</p></NavLink>

            <NavLink onClick={() => setShowMenu(false)} to='/about' ><p className='px-4 py-2 rounded full inline-block'>ABOUT</p></NavLink>

            <NavLink onClick={() => setShowMenu(false)} to='/contact' ><p className='px-4 py-2 rounded full inline-block'>CONTACT</p></NavLink>

          </ul>

        </div>

      </div>

    </div>

  )

}

export default Navbar

**RelatedDoctors.jsx:**

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

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

import { AppContext } from '../context/AppContext'

const RelatedDoctors = ({ speciality, docId }) => {

    const navigate = useNavigate()

    const { doctors } = useContext(AppContext)

    const [relDoc, setRelDoc] = useState([])

    useEffect(() => {

        if (doctors.length > 0 && speciality) {

            const doctorsData = doctors.filter((doc) => doc.speciality === speciality && doc.\_id !== docId)

            setRelDoc(doctorsData)

        }

    }, [doctors, speciality, docId])

    return (

        <div className='flex flex-col items-center gap-4 my-16 text-gray-900'>

            <h1 className='text-3xl font-medium'>Related Doctors</h1>

            <p className='sm:w-1/3 text-center text-sm'>Simply browse through our extensive list of trusted doctors.</p>

            <div className='w-full grid grid-cols-auto gap-4 pt-5 gap-y-6 px-3 sm:px-0'>

                {relDoc.map((item, index) => (

                    <div onClick={() => { navigate(`/appointment/${item.\_id}`); window.scrollTo(0, 0) }} className='border border-blue-200 rounded-xl overflow-hidden cursor-pointer hover:translate-y-[-10px] transition-all duration-500' key={index}>

                        <img className='bg-blue-100' src={item.image} alt="" />

                        <div className='p-4'>

                            <div className='flex items-center gap-2 text-sm text-center text-green-500'>

                                <p className='w-2 h-2 bg-green-500 rounded-full'></p><p>Available</p>

                            </div>

                            <p className='text-gray-900 text-lg font-medium'>{item.name}</p>

                            <p className='text-gray-600 text-sm'>{item.speciality}</p>

                        </div>

                    </div>

                ))}

            </div>

        </div>

    )

}

export default RelatedDoctors

**SpecialityMenu.jsx:**

import React from 'react'

import { specialityData } from '../assets/assets'

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

const SpecialityMenu = () => {

    return (

        <div id='speciality' className='flex flex-col items-center gap-4 py-16 text-gray-800'>

            <h1 className='text-3xl font-medium'>Find by Speciality</h1>

            <p className='sm:w-1/3 text-center text-sm'>Simply browse through our extensive list of trusted doctors, schedule your appointment hassle-free.</p>

            <div className='flex sm:justify-center gap-4 pt-5 w-full overflow-scroll '>

                {specialityData.map((item, index) => (

                    <Link to={`/doctors/${item.speciality}`} onClick={() => scrollTo(0, 0)} className='flex flex-col items-center text-xs cursor-pointer flex-shrink-0 hover:translate-y-[-10px] transition-all duration-500' key={index}>

                        <img className='w-16 sm:w-24 mb-2 ' src={item.image} alt="" />

                        <p>{item.speciality}</p>

                    </Link>

                ))}

            </div>

        </div>

    )

}

export default SpecialityMenu

**TopDoctors.jsx:**

import React, { useContext } from 'react'

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

import { AppContext } from '../context/AppContext'

const TopDoctors = () => {

    const navigate = useNavigate()

    const { doctors } = useContext(AppContext)

    return (

        <div className='flex flex-col items-center gap-4 my-16 text-gray-900 md:mx-10'>

            <h1 className='text-3xl font-medium'>Top Doctors to Book</h1>

            <p className='sm:w-1/3 text-center text-sm'>Simply browse through our extensive list of trusted doctors.</p>

            <div className='w-full grid grid-cols-auto gap-4 pt-5 gap-y-6 px-3 sm:px-0'>

                {doctors.slice(0, 10).map((item, index) => (

                    <div

                        onClick={() => {

                            navigate(`/appointment/${item.\_id}`);

                            window.scrollTo(0, 0)

                        }}

                        className='border border-blue-200 rounded-xl overflow-hidden cursor-pointer hover:translate-y-[-10px] transition-all duration-500'

                        key={index}

                    >

                        <img className='bg-blue-50' src={item.image} alt="" />

                        <div className='p-4'>

                            <div className='flex items-center gap-2 text-sm text-center text-green-500'>

                                <p className='w-2 h-2 bg-green-500 rounded-full'></p><p>Available</p>

                            </div>

                            <p className='text-gray-900 text-lg font-medium'>{item.name}</p>

                            <p className='text-gray-600 text-sm'>{item.speciality}</p>

                        </div>

                    </div>

                ))}

            </div>

            <button onClick={() => { navigate('/doctors'); scrollTo(0, 0) }} className='bg-blue-50 text-gray-600 px-12 py-3 rounded-full mt-10'>more</button>

        </div>

    )

}

export default TopDoctors

**Appointment/src/context:**

**AppContext.jsx:**

import { createContext } from "react";

import { doctors } from "../assets/assets";

export const AppContext = createContext()

const AppContextProvider = (props) => {

    const currencySymbol = '$'

    const value = {

        doctors,

        currencySymbol

    }

    return (

        <AppContext.Provider value={value}>

            {props.children}

        </AppContext.Provider>

    )

}

export default AppContextProvider

**Appointment/src/pages:**

**About.jsx:**

import React from 'react'

import { assets } from '../assets/assets'

const About = () => {

  return (

    <div>

      <div className='text-center text-2xl pt-10 text-gray-500'>

        <p>ABOUT <span className='text-gray-700 font-semibold'>US</span></p>

      </div>

      <div className='my-10 flex flex-col md:flex-row gap-12'>

        <img className='w-full md:max-w-[360px]' src={assets.about\_image} alt="" />

        <div className='flex flex-col justify-center gap-6 md:w-2/4 text-sm text-gray-600'>

          <p>Welcome to Prescripto, your trusted partner in managing your healthcare needs conveniently and efficiently. At Prescripto, we understand the challenges individuals face when it comes to scheduling doctor appointments and managing their health records.</p>

          <p>Prescripto is committed to excellence in healthcare technology. We continuously strive to enhance our platform, integrating the latest advancements to improve user experience and deliver superior service. Whether you're booking your first appointment or managing ongoing care, Prescripto is here to support you every step of the way.</p>

          <b className='text-gray-800'>Our Vision</b>

          <p>Our vision at Prescripto is to create a seamless healthcare experience for every user. We aim to bridge the gap between patients and healthcare providers, making it easier for you to access the care you need, when you need it.</p>

        </div>

      </div>

      <div className='text-xl my-4'>

        <p>WHY  <span className='text-gray-700 font-semibold'>CHOOSE US</span></p>

      </div>

      <div className='flex flex-col md:flex-row mb-20'>

        <div className='border px-10 md:px-16 py-8 sm:py-16 flex flex-col gap-5 text-[15px] hover:bg-primary hover:text-white transition-all duration-300 text-gray-600 cursor-pointer'>

          <b>EFFICIENCY:</b>

          <p>Streamlined appointment scheduling that fits into your busy lifestyle.</p>

        </div>

        <div className='border px-10 md:px-16 py-8 sm:py-16 flex flex-col gap-5 text-[15px] hover:bg-primary hover:text-white transition-all duration-300 text-gray-600 cursor-pointer'>

          <b>CONVENIENCE: </b>

          <p>Access to a network of trusted healthcare professionals in your area.</p>

        </div>

        <div className='border px-10 md:px-16 py-8 sm:py-16 flex flex-col gap-5 text-[15px] hover:bg-primary hover:text-white transition-all duration-300 text-gray-600 cursor-pointer'>

          <b>PERSONALIZATION:</b>

          <p >Tailored recommendations and reminders to help you stay on top of your health.</p>

        </div>

      </div>

    </div>

  )

}

export default About

**Appointment.jsx:**

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

import { useParams } from 'react-router-dom'

import { AppContext } from '../context/AppContext'

import { assets } from '../assets/assets'

import RelatedDoctors from '../components/RelatedDoctors'

const Appointment = () => {

    const { docId } = useParams()

    const { doctors, currencySymbol } = useContext(AppContext)

    const daysOfWeek = ['SUN', 'MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT']

    const [docInfo, setDocInfo] = useState(null)

    const [docSlots, setDocSlots] = useState([])

    const [slotIndex, setSlotIndex] = useState(0)

    const [slotTime, setSlotTime] = useState('')

    const fetchDocInfo = async () => {

        const docInfo = doctors.find((doc) => doc.\_id === docId)

        setDocInfo(docInfo)

    }

    const getAvailableSlots = async () => {

        setDocSlots([])

        // getting current date

        let today = new Date()

        for (let i = 0; i < 7; i++) {

            // getting date with index

            let currentDate = new Date(today)

            currentDate.setDate(today.getDate() + i)

            // setting end time of the date with index

            let endTime = new Date()

            endTime.setDate(today.getDate() + i)

            endTime.setHours(21, 0, 0, 0)

            // setting hours

            if (today.getDate() === currentDate.getDate()) {

                currentDate.setHours(currentDate.getHours() > 10 ? currentDate.getHours() + 1 : 10)

                currentDate.setMinutes(currentDate.getMinutes() > 30 ? 30 : 0)

            } else {

                currentDate.setHours(10)

                currentDate.setMinutes(0)

            }

            let timeSlots = []

            while (currentDate < endTime) {

                let formattedTime = currentDate.toLocaleTimeString([], { hour: '2-digit', minute: '2-digit' })

                // Add slot to array

                timeSlots.push({

                    datetime: new Date(currentDate),

                    time: formattedTime

                })

                // Increment current time by 30 minutes

                currentDate.setMinutes(currentDate.getMinutes() + 30)

            }

            setDocSlots(prev => ([...prev, timeSlots]))

        }

    }

    const bookAppointment = async () => {

        const date = docSlots[slotIndex][0].datetime

        let day = date.getDate()

        let month = date.getMonth() + 1

        let year = date.getFullYear()

        const slotDate = `${day}\_${month}\_${year}`

        console.log(slotDate, slotTime)

    }

    useEffect(() => {

        if (doctors.length > 0) {

            fetchDocInfo()

        }

    }, [doctors, docId])

    useEffect(() => {

        if (docInfo) {

            getAvailableSlots()

        }

    }, [docInfo])

    return docInfo ? (

        <div>

            {/\* ---------- Doctor Details ----------- \*/}

            <div className='flex flex-col sm:flex-row gap-4'>

                <div>

                    <img className='bg-primary w-full sm:max-w-72 rounded-lg' src={docInfo.image} alt="" />

                </div>

                <div className='flex-1 border border-gray-400 rounded-lg p-8 py-7 bg-white mx-2 sm:mx-0 mt-[-80px] sm:mt-0'>

                    {/\* ----- Doc Info : name, degree, experience ----- \*/}

                    <p className='flex items-center gap-2 text-2xl font-medium text-gray-900'>

                        {docInfo.name}

                        <img className='w-5' src={assets.verified\_icon} alt="" />

                    </p>

                    <div className='flex items-center gap-2 text-sm mt-1 text-gray-600'>

                        <p>{docInfo.degree} - {docInfo.speciality}</p>

                        <button className='py-0.5 px-2 border text-xs rounded-full'>{docInfo.experience}</button>

                    </div>

                    {/\* ----- Doc About ----- \*/}

                    <div>

                        <p className='flex items-center gap-1 text-sm font-medium text-gray-900 mt-3'>

                            About <img className='w-3' src={assets.info\_icon} alt="" />

                        </p>

                        <p className='text-sm text-gray-500 max-w-[700px] mt-1'>{docInfo.about}</p>

                    </div>

                    <p className='text-gray-500 font-medium mt-4'>

                        Appointment fee: <span className='text-gray-600'>{currencySymbol}{docInfo.fees}</span>

                    </p>

                </div>

            </div>

            {/\* Booking slots \*/}

            <div className='sm:ml-72 sm:pl-4 mt-4 font-medium text-gray-700'>

                <p>Booking slots</p>

                <div className='flex gap-3 items-center w-full overflow-x-scroll mt-4'>

                    {docSlots.length && docSlots.map((item, index) => (

                        <div

                            onClick={() => setSlotIndex(index)}

                            key={index}

                            className={`text-center py-6 min-w-16 rounded-full cursor-pointer ${slotIndex === index ? 'bg-primary text-white' : 'border border-gray-200'}`}

                        >

                            <p>{item[0] && daysOfWeek[item[0].datetime.getDay()]}</p>

                            <p>{item[0] && item[0].datetime.getDate()}</p>

                        </div>

                    ))}

                </div>

                <div className='flex items-center gap-3 w-full overflow-x-scroll mt-4'>

                    {docSlots.length && docSlots[slotIndex].map((item, index) => (

                        <p

                            onClick={() => setSlotTime(item.time)}

                            key={index}

                            className={`text-sm font-light flex-shrink-0 px-5 py-2 rounded-full cursor-pointer ${item.time === slotTime ? 'bg-primary text-white' : 'text-gray-400 border border-gray-300'}`}

                        >

                            {item.time.toLowerCase()}

                        </p>

                    ))}

                </div>

                <button

                    onClick={bookAppointment}

                    className='bg-primary text-white text-sm font-light px-14 py-3 rounded-full my-6'

                >

                    Book an appointment

                </button>

            </div>

            {/\* Listing Related Doctors \*/}

            <RelatedDoctors speciality={docInfo.speciality} docId={docId} />

        </div>

    ) : null

}

export default Appointment

**Contact.jsx:**

import React from 'react'

import { assets } from '../assets/assets'

const Contact = () => {

  return (

    <div>

      <div className='text-center text-2xl pt-10 text-gray-500'>

        <p>CONTACT <span className='text-gray-700 font-semibold'>US</span></p>

      </div>

      <div className='my-10 flex flex-col justify-center md:flex-row gap-10 mb-28 text-sm'>

        <img className='w-full md:max-w-[360px]' src={assets.contact\_image} alt="" />

        <div className='flex flex-col justify-center items-start gap-6'>

          <p className=' font-semibold text-lg text-gray-600'>OUR OFFICE</p>

          <p className=' text-gray-500'>54709 Willms Station <br /> Suite 350, Washington, USA</p>

          <p className=' text-gray-500'>Tel: (415) 555-0132 <br /> Email: greatstackdev@gmail.com</p>

          <p className=' font-semibold text-lg text-gray-600'>CAREERS AT PRESCRIPTO</p>

          <p className=' text-gray-500'>Learn more about our teams and job openings.</p>

          <button className='border border-black px-8 py-4 text-sm hover:bg-black hover:text-white transition-all duration-500'>Explore Jobs</button>

        </div>

      </div>

    </div>

  )

}

export default Contact

**Doctors.jsx:**

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

import { AppContext } from '../context/AppContext'

import { useNavigate, useParams } from 'react-router-dom'

const Doctors = () => {

  const { speciality } = useParams()

  const [filterDoc, setFilterDoc] = useState([])

  const [showFilter, setShowFilter] = useState(false)

  const navigate = useNavigate();

  const { doctors } = useContext(AppContext)

  const applyFilter = () => {

    if (speciality) {

      setFilterDoc(doctors.filter(doc => doc.speciality === speciality))

    } else {

      setFilterDoc(doctors)

    }

  }

  useEffect(() => {

    applyFilter()

  }, [doctors, speciality])

  return (

    <div>

      <p className='text-gray-600'>Browse through the doctors specialist.</p>

      <div className='flex flex-col sm:flex-row items-start gap-5 mt-5'>

        <button onClick={() => setShowFilter(!showFilter)} className={`py-1 px-3 border rounded text-sm  transition-all sm:hidden ${showFilter ? 'bg-primary text-white' : ''}`}>Filters</button>

        <div className={`flex-col gap-4 text-sm text-gray-600 ${showFilter ? 'flex' : 'hidden sm:flex'}`}>

          <p onClick={() => speciality === 'General Dentist' ? navigate('/doctors') : navigate('/doctors/General Dentist')} className={`w-[94vw] sm:w-auto pl-3 py-1.5 pr-16 border border-gray-300 rounded transition-all cursor-pointer ${speciality === 'General Dentist' ? 'bg-indigo-100 text-black ' : ''}`}>General Dentist</p>

          <p onClick={() => speciality === 'Orthodontist' ? navigate('/doctors') : navigate('/doctors/Orthodontist')} className={`w-[94vw] sm:w-auto pl-3 py-1.5 pr-16 border border-gray-300 rounded transition-all cursor-pointer ${speciality === 'Orthodontist' ? 'bg-indigo-100 text-black ' : ''}`}>Orthodontist</p>

          <p onClick={() => speciality === 'Pediatric Dentist' ? navigate('/doctors') : navigate('/doctors/Pediatric Dentist')} className={`w-[94vw] sm:w-auto pl-3 py-1.5 pr-16 border border-gray-300 rounded transition-all cursor-pointer ${speciality === 'Pediatric Dentist' ? 'bg-indigo-100 text-black ' : ''}`}>Pediatric Dentist</p>

          <p onClick={() => speciality === 'Cosmetic Dentist' ? navigate('/doctors') : navigate('/doctors/Cosmetic Dentist')} className={`w-[94vw] sm:w-auto pl-3 py-1.5 pr-16 border border-gray-300 rounded transition-all cursor-pointer ${speciality === 'Cosmetic Dentist' ? 'bg-indigo-100 text-black ' : ''}`}>Cosmetic Dentist</p>

          </div>

        <div className='w-full grid grid-cols-auto gap-4 gap-y-6'>

          {filterDoc.map((item, index) => (

            <div onClick={() => { navigate(`/appointment/${item.\_id}`); window.scrollTo(0, 0) }} className='border border-indigo-200 rounded-xl overflow-hidden cursor-pointer hover:translate-y-[-10px] transition-all duration-500' key={index}>

              <img className='bg-indigo-50' src={item.image} alt="" />

              <div className='p-4'>

                <div className='flex items-center gap-2 text-sm text-center text-green-500'>

                  <p className='w-2 h-2 bg-green-500 rounded-full'></p><p>Available</p>

                </div>

                <p className='text-neutral-800 text-lg font-medium'>{item.name}</p>

                <p className='text-zinc-600 text-sm'>{item.speciality}</p>

              </div>

            </div>

          ))}

        </div>

      </div>

    </div>

  )

}

export default Doctors

**Home.jsx:**

import React from 'react'

import Header from '../components/Header'

import TopDoctors from '../components/TopDoctors'

import Banner from '../components/Banner'

import SpecialityMenu from '../components/SpecialityMenu'

const Home = () => {

  return (

    <div>

      <Header />

      <SpecialityMenu />

      <TopDoctors />

      <Banner />

    </div>

  )

}

export default Home

**Login.jsx:**

import React, { useState } from 'react'

const Login = () => {

  const [state, setState] = useState('Sign Up')

  const [name, setName] = useState('')

  const [email, setEmail] = useState('')

  const [password, setPassword] = useState('')

  const onSubmitHandler = (event) => {

    event.preventDefault();

  }

  return (

    <form onSubmit={onSubmitHandler} className='min-h-[80vh] flex items-center'>

      <div className='flex flex-col gap-3 m-auto items-start p-8 min-w-[340px] sm:min-w-96 border rounded-xl text-zinc-600 text-sm shadow-lg'>

        <p className='text-2xl font-semibold'>{state === 'Sign Up' ? 'Create Account' : 'Login'}</p>

        <p>Please {state === 'Sign Up' ? 'sign up' : 'log in'} to book appointment</p>

        {state === 'Sign Up'

          ? <div className='w-full '>

            <p>Full Name</p>

            <input onChange={(e) => setName(e.target.value)} value={name} className='border border-zinc-300 rounded w-full p-2 mt-1' type="text" required />

          </div>

          : null

        }

        <div className='w-full '>

          <p>Email</p>

          <input onChange={(e) => setEmail(e.target.value)} value={email} className='border border-zinc-300 rounded w-full p-2 mt-1' type="email" required />

        </div>

        <div className='w-full '>

          <p>Password</p>

          <input onChange={(e) => setPassword(e.target.value)} value={password} className='border border-zinc-300 rounded w-full p-2 mt-1' type="password" required />

        </div>

        <button className='bg-primary text-white w-full py-2 rounded-md text-base'>{state === 'Sign Up' ? 'Create account' : 'Login'}</button>

        {state === 'Sign Up'

          ? <p>Already have an account? <span onClick={() => setState('Login')} className='text-primary underline cursor-pointer'>Login here</span></p>

          : <p>Create an new account? <span onClick={() => setState('Sign Up')} className='text-primary underline cursor-pointer'>Click here</span></p>

        }

      </div>

    </form>

  )

}

export default Login

**MyAppointments.jsx:**

import React, { useContext } from 'react'

import { AppContext } from '../context/AppContext'

const MyAppointments = () => {

    const { doctors } = useContext(AppContext)

    return (

        <div>

            <p className='pb-3 mt-12 font-medium text-zinc-700 border-b'>My appointments</p>

            <div className=''>

                {doctors.slice(0, 1).map((item, index) => (

                    <div key={index} className='grid grid-cols-[1fr\_2fr] gap-4 sm:flex sm:gap-6 py-2 border-b'>

                        <div>

                            <img className='w-32 bg-indigo-50' src={item.image} alt="" />

                        </div>

                        <div className='flex-1 text-sm text-zinc-600'>

                            <p className='text-neutral-800 font-semibold'>{item.name}</p>

                            <p>{item.speciality}</p>

                            {/\* <p className='text-zinc-700 font-medium mt-1'>Address:</p>

                            <p className='text-xs'>{item.address.line1}</p>

                            <p className='text-xs'>{item.address.line2}</p> \*/}

                            <p className='text-xs mt-1'><span className='text-sm text-neutral-700 font-medium'>Date & Time:</span> 02, April, 2025 |  12:00 PM</p>

                        </div>

                        <div></div>

                        <div className='flex flex-col gap-2 justify-end'>

                            <button className='text-sm text-stone-500 text-center sm:min-w-48 py-2 border rounded hover:bg-primary hover:text-white transition-all duration-300'>Pay Online</button>

                            <button className='text-sm text-stone-500 text-center sm:min-w-48 py-2 border rounded hover:bg-red-600 hover:text-white transition-all duration-300'>Cancel appointment</button>

                        </div>

                    </div>

                ))}

            </div>

        </div>

    )

}

export default MyAppointments

**MyProfile:**

import React, { useState } from 'react'

import { assets } from '../assets/assets'

const MyProfile = () => {

    const [isEdit, setIsEdit] = useState(false)

    const [userData, setUserData] = useState({

        name: "Ekta Panchal",

        image: assets.profile\_pic,

        email: 'ekta@gmail.com',

        phone: '+919754367890',

        address: {

            line1: 'Kasarwadavli',

            line2: 'Ghodbunder, Thane',

        },

        gender: 'Female',

        dob: '2003-08-19'

    })

    return (

        <div className='max-w-lg flex flex-col gap-2 text-sm'>

            <img className='w-36 rounded' src={userData.image} alt="" />

            {isEdit

                ? <input className='bg-gray-50 text-3xl font-medium max-w-60 mt-4' type="text" onChange={(e) => setUserData(prev => ({ ...prev, name: e.target.value }))} value={userData.name} />

                : <p className='font-medium text-3xl text-neutral-800 mt-4'>{userData.name}</p>

            }

            <hr className='bg-zinc-400 h-[1px] border-none' />

            <div>

                <p className='text-neutral-500 underline mt-3'>CONTACT INFORMATION</p>

                <div className='grid grid-cols-[1fr\_3fr] gap-y-2.5 mt-3 text-neutral-700'>

                    <p className='font-medium'>Email id:</p>

                    <p className='text-blue-500'>{userData.email}</p>

                    <p className='font-medium'>Phone:</p>

                    {isEdit

                        ? <input className='bg-gray-100 max-w-52' type="text" onChange={(e) => setUserData(prev => ({ ...prev, phone: e.target.value }))} value={userData.phone} />

                        : <p className='text-blue-400'>{userData.phone}</p>}

                    <p className='font-medium'>Address:</p>

                    {isEdit

                        ? <p>

                            <input className='bg-gray-50' type="text" onChange={(e) => setUserData(prev => ({ ...prev, address: { ...prev.address, line1: e.target.value } }))} value={userData.address.line1} />

                            <br />

                            <input className='bg-gray-50' type="text" onChange={(e) => setUserData(prev => ({ ...prev, address: { ...prev.address, line2: e.target.value } }))} value={userData.address.line2} /></p>

                        : <p className='text-gray-500'>{userData.address.line1} <br /> {userData.address.line2}</p>

                    }

                </div>

            </div>

            <div>

                <p className='text-neutral-500 underline mt-3'>BASIC INFORMATION</p>

                <div className='grid grid-cols-[1fr\_3fr] gap-y-2.5 mt-3 text-neutral-700'>

                    <p className='font-medium'>Gender:</p>

                    {isEdit

                        ? <select className='max-w-20 bg-gray-100' onChange={(e) => setUserData(prev => ({ ...prev, gender: e.target.value }))} value={userData.gender} >

                            <option value="Male">Male</option>

                            <option value="Female">Female</option>

                        </select>

                        : <p className='text-gray-400'>{userData.gender}</p>}

                    <p className='font-medium'>Birthday:</p>

                    {isEdit

                        ? <input className='max-w-28 bg-gray-100' type='date' onChange={(e) => setUserData(prev => ({ ...prev, dob: e.target.value }))} value={userData.dob} />

                        : <p className='text-gray-400'>{userData.dob}</p>}

                </div>

            </div>

            <div className='mt-10'>

                {

                    isEdit

                        ? <button onClick={() => setIsEdit(false)} className='border border-primary px-8 py-2 rounded-full hover:bg-primary hover:text-white transition-all'>Save information</button>

                        : <button onClick={() => setIsEdit(true)} className='border border-primary px-8 py-2 rounded-full hover:bg-primary hover:text-white transition-all'>Edit</button>

                }

            </div>

        </div>

    )

}

export default MyProfile

**Appointment/src:**

**App.jsx:**

import React from 'react'

import Navbar from './components/Navbar'

import { Routes, Route } from 'react-router-dom'

import Home from './pages/Home'

import Doctors from './pages/Doctors'

import Login from './pages/Login'

import About from './pages/About'

import Contact from './pages/Contact'

import Appointment from './pages/Appointment'

import MyAppointments from './pages/MyAppointments'

import MyProfile from './pages/MyProfile'

import Footer from './components/Footer'

const App = () => {

  return (

    <div className='mx-4 sm:mx-[10%]'>

      <Navbar />

      <Routes>

        <Route path='/' element={<Doctors />} />

        {/\* <Route path='/doctors' element={<Doctors />} /> \*/}

        <Route path='/doctors/:speciality' element={<Doctors />} />

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

        <Route path='/about' element={<About />} />

        <Route path='/contact' element={<Contact />} />

        <Route path='/appointment/:docId' element={<Appointment />} />

        <Route path='/my-appointments' element={<MyAppointments />} />

        <Route path='/my-profile' element={<MyProfile />} />

      </Routes>

      <Footer />

    </div>

  )

}

export default App

**index.css:**

@import url('https://fonts.googleapis.com/css2?family=Outfit:wght@100..900&display=swap');

@tailwind base;

@tailwind components;

@tailwind utilities;

\* {

    font-family: Outfit;

}

.active hr {

    @apply block

}

@media (max-width:740px) {

    .active  p{

       @apply text-white bg-primary

    }

}

::-webkit-scrollbar {

    @apply hidden

}

**main.jsx:**

import React from 'react'

import ReactDOM from 'react-dom/client'

import App from './App.jsx'

import './index.css'

import { BrowserRouter } from 'react-router-dom'

import AppContextProvider from './context/AppContext.jsx'

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

  <BrowserRouter>

    <AppContextProvider>

      <App />

    </AppContextProvider>

  </BrowserRouter>,

)

**Appointment:**

**.eslintrc.cjs:**

module.exports = {

  root: true,

  env: { browser: true, es2020: true },

  extends: [

    'eslint:recommended',

    'plugin:react/recommended',

    'plugin:react/jsx-runtime',

    'plugin:react-hooks/recommended',

  ],

  ignorePatterns: ['dist', '.eslintrc.cjs'],

  parserOptions: { ecmaVersion: 'latest', sourceType: 'module' },

  settings: { react: { version: '18.2' } },

  plugins: ['react-refresh'],

  rules: {

    'react/jsx-no-target-blank': 'off',

    'react-refresh/only-export-components': [

      'warn',

      { allowConstantExport: true },

    ],

  },

}

**.gitignore:**

# Logs

logs

\*.log

npm-debug.log\*

yarn-debug.log\*

yarn-error.log\*

pnpm-debug.log\*

lerna-debug.log\*

node\_modules

dist

dist-ssr

\*.local

# Editor directories and files

.vscode/\*

!.vscode/extensions.json

.idea

.DS\_Store

\*.suo

\*.ntvs\*

\*.njsproj

\*.sln

\*.sw?

**index.html:**

<!doctype html>

<html lang="en" class="scroll-smooth">

  <head>

    <meta charset="UTF-8" />

    <link rel="icon" href="{{ url\_for('static', filename='assets/favicon.ico') }}" type="image/x-icon">

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Prescripto</title>

  </head>

  <body >

    <div id="root" ></div>

    <script type="module" src="/src/main.jsx"></script>

  </body>

</html>

**package-lock-json:**

{

"name": "frontend",

"version": "0.0.0",

"lockfileVersion": 3,

"requires": true,

"packages": {

"": {

"name": "frontend",

"version": "0.0.0",

"dependencies": {

"axios": "^1.7.3",

"react": "^18.3.1",

"react-dom": "^18.3.1",

"react-router-dom": "^6.26.0",

"react-toastify": "^10.0.5"

},

"devDependencies": {

"@types/react": "^18.3.3",

"@types/react-dom": "^18.3.0",

"@vitejs/plugin-react": "^4.3.1",

"autoprefixer": "^10.4.20",

"eslint": "^8.57.0",

"eslint-plugin-react": "^7.35.0",

"eslint-plugin-react-hooks": "^4.6.2",

"eslint-plugin-react-refresh": "^0.4.9",

"postcss": "^8.4.41",

"tailwindcss": "^3.4.9",

"vite": "^5.4.0"

}

},

"node\_modules/@alloc/quick-lru": {

"version": "5.2.0",

"resolved": "https://registry.npmjs.org/@alloc/quick-lru/-/quick-lru-5.2.0.tgz",

"integrity": "sha512-UrcABB+4bUrFABwbluTIBErXwvbsU/V7TZWfmbgJfbkwiBuziS9gxdODUyuiecfdGQ85jglMW6juS3+z5TsKLw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/@ampproject/remapping": {

"version": "2.3.0",

"resolved": "https://registry.npmjs.org/@ampproject/remapping/-/remapping-2.3.0.tgz",

"integrity": "sha512-30iZtAPgz+LTIYoeivqYo853f02jBYSd5uGnGpkFV0M3xOt9aN73erkgYAmZU43x4VfqcnLxW9Kpg3R5LC4YYw==",

"dev": true,

"license": "Apache-2.0",

"dependencies": {

"@jridgewell/gen-mapping": "^0.3.5",

"@jridgewell/trace-mapping": "^0.3.24"

},

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/@babel/code-frame": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/code-frame/-/code-frame-7.24.7.tgz",

"integrity": "sha512-BcYH1CVJBO9tvyIZ2jVeXgSIMvGZ2FDRvDdOIVQyuklNKSsx+eppDEBq/g47Ayw+RqNFE+URvOShmf+f/qwAlA==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/highlight": "^7.24.7",

"picocolors": "^1.0.0"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/compat-data": {

"version": "7.25.2",

"resolved": "https://registry.npmjs.org/@babel/compat-data/-/compat-data-7.25.2.tgz",

"integrity": "sha512-bYcppcpKBvX4znYaPEeFau03bp89ShqNMLs+rmdptMw+heSZh9+z84d2YG+K7cYLbWwzdjtDoW/uqZmPjulClQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/core": {

"version": "7.25.2",

"resolved": "https://registry.npmjs.org/@babel/core/-/core-7.25.2.tgz",

"integrity": "sha512-BBt3opiCOxUr9euZ5/ro/Xv8/V7yJ5bjYMqG/C1YAo8MIKAnumZalCN+msbci3Pigy4lIQfPUpfMM27HMGaYEA==",

"dev": true,

"license": "MIT",

"dependencies": {

"@ampproject/remapping": "^2.2.0",

"@babel/code-frame": "^7.24.7",

"@babel/generator": "^7.25.0",

"@babel/helper-compilation-targets": "^7.25.2",

"@babel/helper-module-transforms": "^7.25.2",

"@babel/helpers": "^7.25.0",

"@babel/parser": "^7.25.0",

"@babel/template": "^7.25.0",

"@babel/traverse": "^7.25.2",

"@babel/types": "^7.25.2",

"convert-source-map": "^2.0.0",

"debug": "^4.1.0",

"gensync": "^1.0.0-beta.2",

"json5": "^2.2.3",

"semver": "^6.3.1"

},

"engines": {

"node": ">=6.9.0"

},

"funding": {

"type": "opencollective",

"url": "https://opencollective.com/babel"

}

},

"node\_modules/@babel/generator": {

"version": "7.25.0",

"resolved": "https://registry.npmjs.org/@babel/generator/-/generator-7.25.0.tgz",

"integrity": "sha512-3LEEcj3PVW8pW2R1SR1M89g/qrYk/m/mB/tLqn7dn4sbBUQyTqnlod+II2U4dqiGtUmkcnAmkMDralTFZttRiw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/types": "^7.25.0",

"@jridgewell/gen-mapping": "^0.3.5",

"@jridgewell/trace-mapping": "^0.3.25",

"jsesc": "^2.5.1"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-compilation-targets": {

"version": "7.25.2",

"resolved": "https://registry.npmjs.org/@babel/helper-compilation-targets/-/helper-compilation-targets-7.25.2.tgz",

"integrity": "sha512-U2U5LsSaZ7TAt3cfaymQ8WHh0pxvdHoEk6HVpaexxixjyEquMh0L0YNJNM6CTGKMXV1iksi0iZkGw4AcFkPaaw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/compat-data": "^7.25.2",

"@babel/helper-validator-option": "^7.24.8",

"browserslist": "^4.23.1",

"lru-cache": "^5.1.1",

"semver": "^6.3.1"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-module-imports": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/helper-module-imports/-/helper-module-imports-7.24.7.tgz",

"integrity": "sha512-8AyH3C+74cgCVVXow/myrynrAGv+nTVg5vKu2nZph9x7RcRwzmh0VFallJuFTZ9mx6u4eSdXZfcOzSqTUm0HCA==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/traverse": "^7.24.7",

"@babel/types": "^7.24.7"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-module-transforms": {

"version": "7.25.2",

"resolved": "https://registry.npmjs.org/@babel/helper-module-transforms/-/helper-module-transforms-7.25.2.tgz",

"integrity": "sha512-BjyRAbix6j/wv83ftcVJmBt72QtHI56C7JXZoG2xATiLpmoC7dpd8WnkikExHDVPpi/3qCmO6WY1EaXOluiecQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/helper-module-imports": "^7.24.7",

"@babel/helper-simple-access": "^7.24.7",

"@babel/helper-validator-identifier": "^7.24.7",

"@babel/traverse": "^7.25.2"

},

"engines": {

"node": ">=6.9.0"

},

"peerDependencies": {

"@babel/core": "^7.0.0"

}

},

"node\_modules/@babel/helper-plugin-utils": {

"version": "7.24.8",

"resolved": "https://registry.npmjs.org/@babel/helper-plugin-utils/-/helper-plugin-utils-7.24.8.tgz",

"integrity": "sha512-FFWx5142D8h2Mgr/iPVGH5G7w6jDn4jUSpZTyDnQO0Yn7Ks2Kuz6Pci8H6MPCoUJegd/UZQ3tAvfLCxQSnWWwg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-simple-access": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/helper-simple-access/-/helper-simple-access-7.24.7.tgz",

"integrity": "sha512-zBAIvbCMh5Ts+b86r/CjU+4XGYIs+R1j951gxI3KmmxBMhCg4oQMsv6ZXQ64XOm/cvzfU1FmoCyt6+owc5QMYg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/traverse": "^7.24.7",

"@babel/types": "^7.24.7"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-string-parser": {

"version": "7.24.8",

"resolved": "https://registry.npmjs.org/@babel/helper-string-parser/-/helper-string-parser-7.24.8.tgz",

"integrity": "sha512-pO9KhhRcuUyGnJWwyEgnRJTSIZHiT+vMD0kPeD+so0l7mxkMT19g3pjY9GTnHySck/hDzq+dtW/4VgnMkippsQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-validator-identifier": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/helper-validator-identifier/-/helper-validator-identifier-7.24.7.tgz",

"integrity": "sha512-rR+PBcQ1SMQDDyF6X0wxtG8QyLCgUB0eRAGguqRLfkCA87l7yAP7ehq8SNj96OOGTO8OBV70KhuFYcIkHXOg0w==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helper-validator-option": {

"version": "7.24.8",

"resolved": "https://registry.npmjs.org/@babel/helper-validator-option/-/helper-validator-option-7.24.8.tgz",

"integrity": "sha512-xb8t9tD1MHLungh/AIoWYN+gVHaB9kwlu8gffXGSt3FFEIT7RjS+xWbc2vUD1UTZdIpKj/ab3rdqJ7ufngyi2Q==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/helpers": {

"version": "7.25.0",

"resolved": "https://registry.npmjs.org/@babel/helpers/-/helpers-7.25.0.tgz",

"integrity": "sha512-MjgLZ42aCm0oGjJj8CtSM3DB8NOOf8h2l7DCTePJs29u+v7yO/RBX9nShlKMgFnRks/Q4tBAe7Hxnov9VkGwLw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/template": "^7.25.0",

"@babel/types": "^7.25.0"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/highlight": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/highlight/-/highlight-7.24.7.tgz",

"integrity": "sha512-EStJpq4OuY8xYfhGVXngigBJRWxftKX9ksiGDnmlY3o7B/V7KIAc9X4oiK87uPJSc/vs5L869bem5fhZa8caZw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/helper-validator-identifier": "^7.24.7",

"chalk": "^2.4.2",

"js-tokens": "^4.0.0",

"picocolors": "^1.0.0"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/parser": {

"version": "7.25.3",

"resolved": "https://registry.npmjs.org/@babel/parser/-/parser-7.25.3.tgz",

"integrity": "sha512-iLTJKDbJ4hMvFPgQwwsVoxtHyWpKKPBrxkANrSYewDPaPpT5py5yeVkgPIJ7XYXhndxJpaA3PyALSXQ7u8e/Dw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/types": "^7.25.2"

},

"bin": {

"parser": "bin/babel-parser.js"

},

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/@babel/plugin-transform-react-jsx-self": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/plugin-transform-react-jsx-self/-/plugin-transform-react-jsx-self-7.24.7.tgz",

"integrity": "sha512-fOPQYbGSgH0HUp4UJO4sMBFjY6DuWq+2i8rixyUMb3CdGixs/gccURvYOAhajBdKDoGajFr3mUq5rH3phtkGzw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/helper-plugin-utils": "^7.24.7"

},

"engines": {

"node": ">=6.9.0"

},

"peerDependencies": {

"@babel/core": "^7.0.0-0"

}

},

"node\_modules/@babel/plugin-transform-react-jsx-source": {

"version": "7.24.7",

"resolved": "https://registry.npmjs.org/@babel/plugin-transform-react-jsx-source/-/plugin-transform-react-jsx-source-7.24.7.tgz",

"integrity": "sha512-J2z+MWzZHVOemyLweMqngXrgGC42jQ//R0KdxqkIz/OrbVIIlhFI3WigZ5fO+nwFvBlncr4MGapd8vTyc7RPNQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/helper-plugin-utils": "^7.24.7"

},

"engines": {

"node": ">=6.9.0"

},

"peerDependencies": {

"@babel/core": "^7.0.0-0"

}

},

"node\_modules/@babel/template": {

"version": "7.25.0",

"resolved": "https://registry.npmjs.org/@babel/template/-/template-7.25.0.tgz",

"integrity": "sha512-aOOgh1/5XzKvg1jvVz7AVrx2piJ2XBi227DHmbY6y+bM9H2FlN+IfecYu4Xl0cNiiVejlsCri89LUsbj8vJD9Q==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/code-frame": "^7.24.7",

"@babel/parser": "^7.25.0",

"@babel/types": "^7.25.0"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/traverse": {

"version": "7.25.3",

"resolved": "https://registry.npmjs.org/@babel/traverse/-/traverse-7.25.3.tgz",

"integrity": "sha512-HefgyP1x754oGCsKmV5reSmtV7IXj/kpaE1XYY+D9G5PvKKoFfSbiS4M77MdjuwlZKDIKFCffq9rPU+H/s3ZdQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/code-frame": "^7.24.7",

"@babel/generator": "^7.25.0",

"@babel/parser": "^7.25.3",

"@babel/template": "^7.25.0",

"@babel/types": "^7.25.2",

"debug": "^4.3.1",

"globals": "^11.1.0"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@babel/types": {

"version": "7.25.2",

"resolved": "https://registry.npmjs.org/@babel/types/-/types-7.25.2.tgz",

"integrity": "sha512-YTnYtra7W9e6/oAZEHj0bJehPRUlLH9/fbpT5LfB0NhQXyALCRkRs3zH9v07IYhkgpqX6Z78FnuccZr/l4Fs4Q==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/helper-string-parser": "^7.24.8",

"@babel/helper-validator-identifier": "^7.24.7",

"to-fast-properties": "^2.0.0"

},

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/@esbuild/aix-ppc64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/aix-ppc64/-/aix-ppc64-0.21.5.tgz",

"integrity": "sha512-1SDgH6ZSPTlggy1yI6+Dbkiz8xzpHJEVAlF/AM1tHPLsf5STom9rwtjE4hKAF20FfXXNTFqEYXyJNWh1GiZedQ==",

"cpu": [

"ppc64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"aix"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/android-arm": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/android-arm/-/android-arm-0.21.5.tgz",

"integrity": "sha512-vCPvzSjpPHEi1siZdlvAlsPxXl7WbOVUBBAowWug4rJHb68Ox8KualB+1ocNvT5fjv6wpkX6o/iEpbDrf68zcg==",

"cpu": [

"arm"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"android"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/android-arm64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/android-arm64/-/android-arm64-0.21.5.tgz",

"integrity": "sha512-c0uX9VAUBQ7dTDCjq+wdyGLowMdtR/GoC2U5IYk/7D1H1JYC0qseD7+11iMP2mRLN9RcCMRcjC4YMclCzGwS/A==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"android"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/android-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/android-x64/-/android-x64-0.21.5.tgz",

"integrity": "sha512-D7aPRUUNHRBwHxzxRvp856rjUHRFW1SdQATKXH2hqA0kAZb1hKmi02OpYRacl0TxIGz/ZmXWlbZgjwWYaCakTA==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"android"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/darwin-arm64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/darwin-arm64/-/darwin-arm64-0.21.5.tgz",

"integrity": "sha512-DwqXqZyuk5AiWWf3UfLiRDJ5EDd49zg6O9wclZ7kUMv2WRFr4HKjXp/5t8JZ11QbQfUS6/cRCKGwYhtNAY88kQ==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"darwin"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/darwin-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/darwin-x64/-/darwin-x64-0.21.5.tgz",

"integrity": "sha512-se/JjF8NlmKVG4kNIuyWMV/22ZaerB+qaSi5MdrXtd6R08kvs2qCN4C09miupktDitvh8jRFflwGFBQcxZRjbw==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"darwin"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/freebsd-arm64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/freebsd-arm64/-/freebsd-arm64-0.21.5.tgz",

"integrity": "sha512-5JcRxxRDUJLX8JXp/wcBCy3pENnCgBR9bN6JsY4OmhfUtIHe3ZW0mawA7+RDAcMLrMIZaf03NlQiX9DGyB8h4g==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"freebsd"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/freebsd-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/freebsd-x64/-/freebsd-x64-0.21.5.tgz",

"integrity": "sha512-J95kNBj1zkbMXtHVH29bBriQygMXqoVQOQYA+ISs0/2l3T9/kj42ow2mpqerRBxDJnmkUDCaQT/dfNXWX/ZZCQ==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"freebsd"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-arm": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-arm/-/linux-arm-0.21.5.tgz",

"integrity": "sha512-bPb5AHZtbeNGjCKVZ9UGqGwo8EUu4cLq68E95A53KlxAPRmUyYv2D6F0uUI65XisGOL1hBP5mTronbgo+0bFcA==",

"cpu": [

"arm"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-arm64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-arm64/-/linux-arm64-0.21.5.tgz",

"integrity": "sha512-ibKvmyYzKsBeX8d8I7MH/TMfWDXBF3db4qM6sy+7re0YXya+K1cem3on9XgdT2EQGMu4hQyZhan7TeQ8XkGp4Q==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-ia32": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-ia32/-/linux-ia32-0.21.5.tgz",

"integrity": "sha512-YvjXDqLRqPDl2dvRODYmmhz4rPeVKYvppfGYKSNGdyZkA01046pLWyRKKI3ax8fbJoK5QbxblURkwK/MWY18Tg==",

"cpu": [

"ia32"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-loong64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-loong64/-/linux-loong64-0.21.5.tgz",

"integrity": "sha512-uHf1BmMG8qEvzdrzAqg2SIG/02+4/DHB6a9Kbya0XDvwDEKCoC8ZRWI5JJvNdUjtciBGFQ5PuBlpEOXQj+JQSg==",

"cpu": [

"loong64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-mips64el": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-mips64el/-/linux-mips64el-0.21.5.tgz",

"integrity": "sha512-IajOmO+KJK23bj52dFSNCMsz1QP1DqM6cwLUv3W1QwyxkyIWecfafnI555fvSGqEKwjMXVLokcV5ygHW5b3Jbg==",

"cpu": [

"mips64el"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-ppc64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-ppc64/-/linux-ppc64-0.21.5.tgz",

"integrity": "sha512-1hHV/Z4OEfMwpLO8rp7CvlhBDnjsC3CttJXIhBi+5Aj5r+MBvy4egg7wCbe//hSsT+RvDAG7s81tAvpL2XAE4w==",

"cpu": [

"ppc64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-riscv64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-riscv64/-/linux-riscv64-0.21.5.tgz",

"integrity": "sha512-2HdXDMd9GMgTGrPWnJzP2ALSokE/0O5HhTUvWIbD3YdjME8JwvSCnNGBnTThKGEB91OZhzrJ4qIIxk/SBmyDDA==",

"cpu": [

"riscv64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-s390x": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-s390x/-/linux-s390x-0.21.5.tgz",

"integrity": "sha512-zus5sxzqBJD3eXxwvjN1yQkRepANgxE9lgOW2qLnmr8ikMTphkjgXu1HR01K4FJg8h1kEEDAqDcZQtbrRnB41A==",

"cpu": [

"s390x"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/linux-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/linux-x64/-/linux-x64-0.21.5.tgz",

"integrity": "sha512-1rYdTpyv03iycF1+BhzrzQJCdOuAOtaqHTWJZCWvijKD2N5Xu0TtVC8/+1faWqcP9iBCWOmjmhoH94dH82BxPQ==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/netbsd-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/netbsd-x64/-/netbsd-x64-0.21.5.tgz",

"integrity": "sha512-Woi2MXzXjMULccIwMnLciyZH4nCIMpWQAs049KEeMvOcNADVxo0UBIQPfSmxB3CWKedngg7sWZdLvLczpe0tLg==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"netbsd"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/openbsd-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/openbsd-x64/-/openbsd-x64-0.21.5.tgz",

"integrity": "sha512-HLNNw99xsvx12lFBUwoT8EVCsSvRNDVxNpjZ7bPn947b8gJPzeHWyNVhFsaerc0n3TsbOINvRP2byTZ5LKezow==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"openbsd"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/sunos-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/sunos-x64/-/sunos-x64-0.21.5.tgz",

"integrity": "sha512-6+gjmFpfy0BHU5Tpptkuh8+uw3mnrvgs+dSPQXQOv3ekbordwnzTVEb4qnIvQcYXq6gzkyTnoZ9dZG+D4garKg==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"sunos"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/win32-arm64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/win32-arm64/-/win32-arm64-0.21.5.tgz",

"integrity": "sha512-Z0gOTd75VvXqyq7nsl93zwahcTROgqvuAcYDUr+vOv8uHhNSKROyU961kgtCD1e95IqPKSQKH7tBTslnS3tA8A==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"win32"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/win32-ia32": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/win32-ia32/-/win32-ia32-0.21.5.tgz",

"integrity": "sha512-SWXFF1CL2RVNMaVs+BBClwtfZSvDgtL//G/smwAc5oVK/UPu2Gu9tIaRgFmYFFKrmg3SyAjSrElf0TiJ1v8fYA==",

"cpu": [

"ia32"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"win32"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@esbuild/win32-x64": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/@esbuild/win32-x64/-/win32-x64-0.21.5.tgz",

"integrity": "sha512-tQd/1efJuzPC6rCFwEvLtci/xNFcTZknmXs98FYDfGE4wP9ClFV98nyKrzJKVPMhdDnjzLhdUyMX4PsQAPjwIw==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"win32"

],

"engines": {

"node": ">=12"

}

},

"node\_modules/@eslint-community/eslint-utils": {

"version": "4.4.0",

"resolved": "https://registry.npmjs.org/@eslint-community/eslint-utils/-/eslint-utils-4.4.0.tgz",

"integrity": "sha512-1/sA4dwrzBAyeUoQ6oxahHKmrZvsnLCg4RfxW3ZFGGmQkSNQPFNLV9CUEFQP1x9EYXHTo5p6xdhZM1Ne9p/AfA==",

"dev": true,

"license": "MIT",

"dependencies": {

"eslint-visitor-keys": "^3.3.0"

},

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

},

"peerDependencies": {

"eslint": "^6.0.0 || ^7.0.0 || >=8.0.0"

}

},

"node\_modules/@eslint-community/regexpp": {

"version": "4.11.0",

"resolved": "https://registry.npmjs.org/@eslint-community/regexpp/-/regexpp-4.11.0.tgz",

"integrity": "sha512-G/M/tIiMrTAxEWRfLfQJMmGNX28IxBg4PBz8XqQhqUHLFI6TL2htpIB1iQCj144V5ee/JaKyT9/WZ0MGZWfA7A==",

"dev": true,

"license": "MIT",

"engines": {

"node": "^12.0.0 || ^14.0.0 || >=16.0.0"

}

},

"node\_modules/@eslint/eslintrc": {

"version": "2.1.4",

"resolved": "https://registry.npmjs.org/@eslint/eslintrc/-/eslintrc-2.1.4.tgz",

"integrity": "sha512-269Z39MS6wVJtsoUl10L60WdkhJVdPG24Q4eZTH3nnF6lpvSShEK3wQjDX9JRWAUPvPh7COouPpU9IrqaZFvtQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"ajv": "^6.12.4",

"debug": "^4.3.2",

"espree": "^9.6.0",

"globals": "^13.19.0",

"ignore": "^5.2.0",

"import-fresh": "^3.2.1",

"js-yaml": "^4.1.0",

"minimatch": "^3.1.2",

"strip-json-comments": "^3.1.1"

},

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

},

"funding": {

"url": "https://opencollective.com/eslint"

}

},

"node\_modules/@eslint/eslintrc/node\_modules/globals": {

"version": "13.24.0",

"resolved": "https://registry.npmjs.org/globals/-/globals-13.24.0.tgz",

"integrity": "sha512-AhO5QUcj8llrbG09iWhPU2B204J1xnPeL8kQmVorSsy+Sjj1sk8gIyh6cUocGmH4L0UuhAJy+hJMRA4mgA4mFQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"type-fest": "^0.20.2"

},

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/@eslint/js": {

"version": "8.57.0",

"resolved": "https://registry.npmjs.org/@eslint/js/-/js-8.57.0.tgz",

"integrity": "sha512-Ys+3g2TaW7gADOJzPt83SJtCDhMjndcDMFVQ/Tj9iA1BfJzFKD9mAUXT3OenpuPHbI6P/myECxRJrofUsDx/5g==",

"dev": true,

"license": "MIT",

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

}

},

"node\_modules/@humanwhocodes/config-array": {

"version": "0.11.14",

"resolved": "https://registry.npmjs.org/@humanwhocodes/config-array/-/config-array-0.11.14.tgz",

"integrity": "sha512-3T8LkOmg45BV5FICb15QQMsyUSWrQ8AygVfC7ZG32zOalnqrilm018ZVCw0eapXux8FtA33q8PSRSstjee3jSg==",

"deprecated": "Use @eslint/config-array instead",

"dev": true,

"license": "Apache-2.0",

"dependencies": {

"@humanwhocodes/object-schema": "^2.0.2",

"debug": "^4.3.1",

"minimatch": "^3.0.5"

},

"engines": {

"node": ">=10.10.0"

}

},

"node\_modules/@humanwhocodes/module-importer": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/@humanwhocodes/module-importer/-/module-importer-1.0.1.tgz",

"integrity": "sha512-bxveV4V8v5Yb4ncFTT3rPSgZBOpCkjfK0y4oVVVJwIuDVBRMDXrPyXRL988i5ap9m9bnyEEjWfm5WkBmtffLfA==",

"dev": true,

"license": "Apache-2.0",

"engines": {

"node": ">=12.22"

},

"funding": {

"type": "github",

"url": "https://github.com/sponsors/nzakas"

}

},

"node\_modules/@humanwhocodes/object-schema": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/@humanwhocodes/object-schema/-/object-schema-2.0.3.tgz",

"integrity": "sha512-93zYdMES/c1D69yZiKDBj0V24vqNzB/koF26KPaagAfd3P/4gUlh3Dys5ogAK+Exi9QyzlD8x/08Zt7wIKcDcA==",

"deprecated": "Use @eslint/object-schema instead",

"dev": true,

"license": "BSD-3-Clause"

},

"node\_modules/@isaacs/cliui": {

"version": "8.0.2",

"resolved": "https://registry.npmjs.org/@isaacs/cliui/-/cliui-8.0.2.tgz",

"integrity": "sha512-O8jcjabXaleOG9DQ0+ARXWZBTfnP4WNAqzuiJK7ll44AmxGKv/J2M4TPjxjY3znBCfvBXFzucm1twdyFybFqEA==",

"dev": true,

"license": "ISC",

"dependencies": {

"string-width": "^5.1.2",

"string-width-cjs": "npm:string-width@^4.2.0",

"strip-ansi": "^7.0.1",

"strip-ansi-cjs": "npm:strip-ansi@^6.0.1",

"wrap-ansi": "^8.1.0",

"wrap-ansi-cjs": "npm:wrap-ansi@^7.0.0"

},

"engines": {

"node": ">=12"

}

},

"node\_modules/@isaacs/cliui/node\_modules/ansi-regex": {

"version": "6.0.1",

"resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-6.0.1.tgz",

"integrity": "sha512-n5M855fKb2SsfMIiFFoVrABHJC8QtHwVx+mHWP3QcEqBHYienj5dHSgjbxtC0WEZXYt4wcD6zrQElDPhFuZgfA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/ansi-regex?sponsor=1"

}

},

"node\_modules/@isaacs/cliui/node\_modules/strip-ansi": {

"version": "7.1.0",

"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-7.1.0.tgz",

"integrity": "sha512-iq6eVVI64nQQTRYq2KtEg2d2uU7LElhTJwsH4YzIHZshxlgZms/wIc4VoDQTlG/IvVIrBKG06CrZnp0qv7hkcQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-regex": "^6.0.1"

},

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/strip-ansi?sponsor=1"

}

},

"node\_modules/@jridgewell/gen-mapping": {

"version": "0.3.5",

"resolved": "https://registry.npmjs.org/@jridgewell/gen-mapping/-/gen-mapping-0.3.5.tgz",

"integrity": "sha512-IzL8ZoEDIBRWEzlCcRhOaCupYyN5gdIK+Q6fbFdPDg6HqX6jpkItn7DFIpW9LQzXG6Df9sA7+OKnq0qlz/GaQg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@jridgewell/set-array": "^1.2.1",

"@jridgewell/sourcemap-codec": "^1.4.10",

"@jridgewell/trace-mapping": "^0.3.24"

},

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/@jridgewell/resolve-uri": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/@jridgewell/resolve-uri/-/resolve-uri-3.1.2.tgz",

"integrity": "sha512-bRISgCIjP20/tbWSPWMEi54QVPRZExkuD9lJL+UIxUKtwVJA8wW1Trb1jMs1RFXo1CBTNZ/5hpC9QvmKWdopKw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/@jridgewell/set-array": {

"version": "1.2.1",

"resolved": "https://registry.npmjs.org/@jridgewell/set-array/-/set-array-1.2.1.tgz",

"integrity": "sha512-R8gLRTZeyp03ymzP/6Lil/28tGeGEzhx1q2k703KGWRAI1VdvPIXdG70VJc2pAMw3NA6JKL5hhFu1sJX0Mnn/A==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/@jridgewell/sourcemap-codec": {

"version": "1.5.0",

"resolved": "https://registry.npmjs.org/@jridgewell/sourcemap-codec/-/sourcemap-codec-1.5.0.tgz",

"integrity": "sha512-gv3ZRaISU3fjPAgNsriBRqGWQL6quFx04YMPW/zD8XMLsU32mhCCbfbO6KZFLjvYpCZ8zyDEgqsgf+PwPaM7GQ==",

"dev": true,

"license": "MIT"

},

"node\_modules/@jridgewell/trace-mapping": {

"version": "0.3.25",

"resolved": "https://registry.npmjs.org/@jridgewell/trace-mapping/-/trace-mapping-0.3.25.tgz",

"integrity": "sha512-vNk6aEwybGtawWmy/PzwnGDOjCkLWSD2wqvjGGAgOAwCGWySYXfYoxt00IJkTF+8Lb57DwOb3Aa0o9CApepiYQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"@jridgewell/resolve-uri": "^3.1.0",

"@jridgewell/sourcemap-codec": "^1.4.14"

}

},

"node\_modules/@nodelib/fs.scandir": {

"version": "2.1.5",

"resolved": "https://registry.npmjs.org/@nodelib/fs.scandir/-/fs.scandir-2.1.5.tgz",

"integrity": "sha512-vq24Bq3ym5HEQm2NKCr3yXDwjc7vTsEThRDnkp2DK9p1uqLR+DHurm/NOTo0KG7HYHU7eppKZj3MyqYuMBf62g==",

"dev": true,

"license": "MIT",

"dependencies": {

"@nodelib/fs.stat": "2.0.5",

"run-parallel": "^1.1.9"

},

"engines": {

"node": ">= 8"

}

},

"node\_modules/@nodelib/fs.stat": {

"version": "2.0.5",

"resolved": "https://registry.npmjs.org/@nodelib/fs.stat/-/fs.stat-2.0.5.tgz",

"integrity": "sha512-RkhPPp2zrqDAQA/2jNhnztcPAlv64XdhIp7a7454A5ovI7Bukxgt7MX7udwAu3zg1DcpPU0rz3VV1SeaqvY4+A==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 8"

}

},

"node\_modules/@nodelib/fs.walk": {

"version": "1.2.8",

"resolved": "https://registry.npmjs.org/@nodelib/fs.walk/-/fs.walk-1.2.8.tgz",

"integrity": "sha512-oGB+UxlgWcgQkgwo8GcEGwemoTFt3FIO9ababBmaGwXIoBKZ+GTy0pP185beGg7Llih/NSHSV2XAs1lnznocSg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@nodelib/fs.scandir": "2.1.5",

"fastq": "^1.6.0"

},

"engines": {

"node": ">= 8"

}

},

"node\_modules/@pkgjs/parseargs": {

"version": "0.11.0",

"resolved": "https://registry.npmjs.org/@pkgjs/parseargs/-/parseargs-0.11.0.tgz",

"integrity": "sha512-+1VkjdD0QBLPodGrJUeqarH8VAIvQODIbwh9XpP5Syisf7YoQgsJKPNFoqqLQlu+VQ/tVSshMR6loPMn8U+dPg==",

"dev": true,

"license": "MIT",

"optional": true,

"engines": {

"node": ">=14"

}

},

"node\_modules/@remix-run/router": {

"version": "1.19.0",

"resolved": "https://registry.npmjs.org/@remix-run/router/-/router-1.19.0.tgz",

"integrity": "sha512-zDICCLKEwbVYTS6TjYaWtHXxkdoUvD/QXvyVZjGCsWz5vyH7aFeONlPffPdW+Y/t6KT0MgXb2Mfjun9YpWN1dA==",

"license": "MIT",

"engines": {

"node": ">=14.0.0"

}

},

"node\_modules/@rollup/rollup-android-arm-eabi": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-android-arm-eabi/-/rollup-android-arm-eabi-4.20.0.tgz",

"integrity": "sha512-TSpWzflCc4VGAUJZlPpgAJE1+V60MePDQnBd7PPkpuEmOy8i87aL6tinFGKBFKuEDikYpig72QzdT3QPYIi+oA==",

"cpu": [

"arm"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"android"

]

},

"node\_modules/@rollup/rollup-android-arm64": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-android-arm64/-/rollup-android-arm64-4.20.0.tgz",

"integrity": "sha512-u00Ro/nok7oGzVuh/FMYfNoGqxU5CPWz1mxV85S2w9LxHR8OoMQBuSk+3BKVIDYgkpeOET5yXkx90OYFc+ytpQ==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"android"

]

},

"node\_modules/@rollup/rollup-darwin-arm64": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-darwin-arm64/-/rollup-darwin-arm64-4.20.0.tgz",

"integrity": "sha512-uFVfvzvsdGtlSLuL0ZlvPJvl6ZmrH4CBwLGEFPe7hUmf7htGAN+aXo43R/V6LATyxlKVC/m6UsLb7jbG+LG39Q==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"darwin"

]

},

"node\_modules/@rollup/rollup-darwin-x64": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-darwin-x64/-/rollup-darwin-x64-4.20.0.tgz",

"integrity": "sha512-xbrMDdlev53vNXexEa6l0LffojxhqDTBeL+VUxuuIXys4x6xyvbKq5XqTXBCEUA8ty8iEJblHvFaWRJTk/icAQ==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"darwin"

]

},

"node\_modules/@rollup/rollup-linux-arm-gnueabihf": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm-gnueabihf/-/rollup-linux-arm-gnueabihf-4.20.0.tgz",

"integrity": "sha512-jMYvxZwGmoHFBTbr12Xc6wOdc2xA5tF5F2q6t7Rcfab68TT0n+r7dgawD4qhPEvasDsVpQi+MgDzj2faOLsZjA==",

"cpu": [

"arm"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-arm-musleabihf": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm-musleabihf/-/rollup-linux-arm-musleabihf-4.20.0.tgz",

"integrity": "sha512-1asSTl4HKuIHIB1GcdFHNNZhxAYEdqML/MW4QmPS4G0ivbEcBr1JKlFLKsIRqjSwOBkdItn3/ZDlyvZ/N6KPlw==",

"cpu": [

"arm"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-arm64-gnu": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm64-gnu/-/rollup-linux-arm64-gnu-4.20.0.tgz",

"integrity": "sha512-COBb8Bkx56KldOYJfMf6wKeYJrtJ9vEgBRAOkfw6Ens0tnmzPqvlpjZiLgkhg6cA3DGzCmLmmd319pmHvKWWlQ==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-arm64-musl": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm64-musl/-/rollup-linux-arm64-musl-4.20.0.tgz",

"integrity": "sha512-+it+mBSyMslVQa8wSPvBx53fYuZK/oLTu5RJoXogjk6x7Q7sz1GNRsXWjn6SwyJm8E/oMjNVwPhmNdIjwP135Q==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-powerpc64le-gnu": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-powerpc64le-gnu/-/rollup-linux-powerpc64le-gnu-4.20.0.tgz",

"integrity": "sha512-yAMvqhPfGKsAxHN8I4+jE0CpLWD8cv4z7CK7BMmhjDuz606Q2tFKkWRY8bHR9JQXYcoLfopo5TTqzxgPUjUMfw==",

"cpu": [

"ppc64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-riscv64-gnu": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-riscv64-gnu/-/rollup-linux-riscv64-gnu-4.20.0.tgz",

"integrity": "sha512-qmuxFpfmi/2SUkAw95TtNq/w/I7Gpjurx609OOOV7U4vhvUhBcftcmXwl3rqAek+ADBwSjIC4IVNLiszoj3dPA==",

"cpu": [

"riscv64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-s390x-gnu": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-s390x-gnu/-/rollup-linux-s390x-gnu-4.20.0.tgz",

"integrity": "sha512-I0BtGXddHSHjV1mqTNkgUZLnS3WtsqebAXv11D5BZE/gfw5KoyXSAXVqyJximQXNvNzUo4GKlCK/dIwXlz+jlg==",

"cpu": [

"s390x"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-x64-gnu": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-x64-gnu/-/rollup-linux-x64-gnu-4.20.0.tgz",

"integrity": "sha512-y+eoL2I3iphUg9tN9GB6ku1FA8kOfmF4oUEWhztDJ4KXJy1agk/9+pejOuZkNFhRwHAOxMsBPLbXPd6mJiCwew==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-linux-x64-musl": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-linux-x64-musl/-/rollup-linux-x64-musl-4.20.0.tgz",

"integrity": "sha512-hM3nhW40kBNYUkZb/r9k2FKK+/MnKglX7UYd4ZUy5DJs8/sMsIbqWK2piZtVGE3kcXVNj3B2IrUYROJMMCikNg==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"linux"

]

},

"node\_modules/@rollup/rollup-win32-arm64-msvc": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-win32-arm64-msvc/-/rollup-win32-arm64-msvc-4.20.0.tgz",

"integrity": "sha512-psegMvP+Ik/Bg7QRJbv8w8PAytPA7Uo8fpFjXyCRHWm6Nt42L+JtoqH8eDQ5hRP7/XW2UiIriy1Z46jf0Oa1kA==",

"cpu": [

"arm64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"win32"

]

},

"node\_modules/@rollup/rollup-win32-ia32-msvc": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-win32-ia32-msvc/-/rollup-win32-ia32-msvc-4.20.0.tgz",

"integrity": "sha512-GabekH3w4lgAJpVxkk7hUzUf2hICSQO0a/BLFA11/RMxQT92MabKAqyubzDZmMOC/hcJNlc+rrypzNzYl4Dx7A==",

"cpu": [

"ia32"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"win32"

]

},

"node\_modules/@rollup/rollup-win32-x64-msvc": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/@rollup/rollup-win32-x64-msvc/-/rollup-win32-x64-msvc-4.20.0.tgz",

"integrity": "sha512-aJ1EJSuTdGnM6qbVC4B5DSmozPTqIag9fSzXRNNo+humQLG89XpPgdt16Ia56ORD7s+H8Pmyx44uczDQ0yDzpg==",

"cpu": [

"x64"

],

"dev": true,

"license": "MIT",

"optional": true,

"os": [

"win32"

]

},

"node\_modules/@types/babel\_\_core": {

"version": "7.20.5",

"resolved": "https://registry.npmjs.org/@types/babel\_\_core/-/babel\_\_core-7.20.5.tgz",

"integrity": "sha512-qoQprZvz5wQFJwMDqeseRXWv3rqMvhgpbXFfVyWhbx9X47POIA6i/+dXefEmZKoAgOaTdaIgNSMqMIU61yRyzA==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/parser": "^7.20.7",

"@babel/types": "^7.20.7",

"@types/babel\_\_generator": "\*",

"@types/babel\_\_template": "\*",

"@types/babel\_\_traverse": "\*"

}

},

"node\_modules/@types/babel\_\_generator": {

"version": "7.6.8",

"resolved": "https://registry.npmjs.org/@types/babel\_\_generator/-/babel\_\_generator-7.6.8.tgz",

"integrity": "sha512-ASsj+tpEDsEiFr1arWrlN6V3mdfjRMZt6LtK/Vp/kreFLnr5QH5+DhvD5nINYZXzwJvXeGq+05iUXcAzVrqWtw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/types": "^7.0.0"

}

},

"node\_modules/@types/babel\_\_template": {

"version": "7.4.4",

"resolved": "https://registry.npmjs.org/@types/babel\_\_template/-/babel\_\_template-7.4.4.tgz",

"integrity": "sha512-h/NUaSyG5EyxBIp8YRxo4RMe2/qQgvyowRwVMzhYhBCONbW8PUsg4lkFMrhgZhUe5z3L3MiLDuvyJ/CaPa2A8A==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/parser": "^7.1.0",

"@babel/types": "^7.0.0"

}

},

"node\_modules/@types/babel\_\_traverse": {

"version": "7.20.6",

"resolved": "https://registry.npmjs.org/@types/babel\_\_traverse/-/babel\_\_traverse-7.20.6.tgz",

"integrity": "sha512-r1bzfrm0tomOI8g1SzvCaQHo6Lcv6zu0EA+W2kHrt8dyrHQxGzBBL4kdkzIS+jBMV+EYcMAEAqXqYaLJq5rOZg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/types": "^7.20.7"

}

},

"node\_modules/@types/estree": {

"version": "1.0.5",

"resolved": "https://registry.npmjs.org/@types/estree/-/estree-1.0.5.tgz",

"integrity": "sha512-/kYRxGDLWzHOB7q+wtSUQlFrtcdUccpfy+X+9iMBpHK8QLLhx2wIPYuS5DYtR9Wa/YlZAbIovy7qVdB1Aq6Lyw==",

"dev": true,

"license": "MIT"

},

"node\_modules/@types/prop-types": {

"version": "15.7.12",

"resolved": "https://registry.npmjs.org/@types/prop-types/-/prop-types-15.7.12.tgz",

"integrity": "sha512-5zvhXYtRNRluoE/jAp4GVsSduVUzNWKkOZrCDBWYtE7biZywwdC2AcEzg+cSMLFRfVgeAFqpfNabiPjxFddV1Q==",

"dev": true,

"license": "MIT"

},

"node\_modules/@types/react": {

"version": "18.3.3",

"resolved": "https://registry.npmjs.org/@types/react/-/react-18.3.3.tgz",

"integrity": "sha512-hti/R0pS0q1/xx+TsI73XIqk26eBsISZ2R0wUijXIngRK9R/e7Xw/cXVxQK7R5JjW+SV4zGcn5hXjudkN/pLIw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@types/prop-types": "\*",

"csstype": "^3.0.2"

}

},

"node\_modules/@types/react-dom": {

"version": "18.3.0",

"resolved": "https://registry.npmjs.org/@types/react-dom/-/react-dom-18.3.0.tgz",

"integrity": "sha512-EhwApuTmMBmXuFOikhQLIBUn6uFg81SwLMOAUgodJF14SOBOCMdU04gDoYi0WOJJHD144TL32z4yDqCW3dnkQg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@types/react": "\*"

}

},

"node\_modules/@ungap/structured-clone": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/@ungap/structured-clone/-/structured-clone-1.2.0.tgz",

"integrity": "sha512-zuVdFrMJiuCDQUMCzQaD6KL28MjnqqN8XnAqiEq9PNm/hCPTSGfrXCOfwj1ow4LFb/tNymJPwsNbVePc1xFqrQ==",

"dev": true,

"license": "ISC"

},

"node\_modules/@vitejs/plugin-react": {

"version": "4.3.1",

"resolved": "https://registry.npmjs.org/@vitejs/plugin-react/-/plugin-react-4.3.1.tgz",

"integrity": "sha512-m/V2syj5CuVnaxcUJOQRel/Wr31FFXRFlnOoq1TVtkCxsY5veGMTEmpWHndrhB2U8ScHtCQB1e+4hWYExQc6Lg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@babel/core": "^7.24.5",

"@babel/plugin-transform-react-jsx-self": "^7.24.5",

"@babel/plugin-transform-react-jsx-source": "^7.24.1",

"@types/babel\_\_core": "^7.20.5",

"react-refresh": "^0.14.2"

},

"engines": {

"node": "^14.18.0 || >=16.0.0"

},

"peerDependencies": {

"vite": "^4.2.0 || ^5.0.0"

}

},

"node\_modules/acorn": {

"version": "8.12.1",

"resolved": "https://registry.npmjs.org/acorn/-/acorn-8.12.1.tgz",

"integrity": "sha512-tcpGyI9zbizT9JbV6oYE477V6mTlXvvi0T0G3SNIYE2apm/G5huBa1+K89VGeovbg+jycCrfhl3ADxErOuO6Jg==",

"dev": true,

"license": "MIT",

"bin": {

"acorn": "bin/acorn"

},

"engines": {

"node": ">=0.4.0"

}

},

"node\_modules/acorn-jsx": {

"version": "5.3.2",

"resolved": "https://registry.npmjs.org/acorn-jsx/-/acorn-jsx-5.3.2.tgz",

"integrity": "sha512-rq9s+JNhf0IChjtDXxllJ7g41oZk5SlXtp0LHwyA5cejwn7vKmKp4pPri6YEePv2PU65sAsegbXtIinmDFDXgQ==",

"dev": true,

"license": "MIT",

"peerDependencies": {

"acorn": "^6.0.0 || ^7.0.0 || ^8.0.0"

}

},

"node\_modules/ajv": {

"version": "6.12.6",

"resolved": "https://registry.npmjs.org/ajv/-/ajv-6.12.6.tgz",

"integrity": "sha512-j3fVLgvTo527anyYyJOGTYJbG+vnnQYvE0m5mmkc1TK+nxAppkCLMIL0aZ4dblVCNoGShhm+kzE4ZUykBoMg4g==",

"dev": true,

"license": "MIT",

"dependencies": {

"fast-deep-equal": "^3.1.1",

"fast-json-stable-stringify": "^2.0.0",

"json-schema-traverse": "^0.4.1",

"uri-js": "^4.2.2"

},

"funding": {

"type": "github",

"url": "https://github.com/sponsors/epoberezkin"

}

},

"node\_modules/ansi-regex": {

"version": "5.0.1",

"resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz",

"integrity": "sha512-quJQXlTSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMlMWv37qOOb9pdJlMUEKFQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/ansi-styles": {

"version": "3.2.1",

"resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-3.2.1.tgz",

"integrity": "sha512-VT0ZI6kZRdTh8YyJw3SMbYm/u+NqfsAxEpWO0Pf9sq8/e94WxxOpPKx9FR1FlyCtOVDNOQ+8ntlqFxiRc+r5qA==",

"dev": true,

"license": "MIT",

"dependencies": {

"color-convert": "^1.9.0"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/any-promise": {

"version": "1.3.0",

"resolved": "https://registry.npmjs.org/any-promise/-/any-promise-1.3.0.tgz",

"integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFIegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3IqX9OB4MZR1b2A==",

"dev": true,

"license": "MIT"

},

"node\_modules/anymatch": {

"version": "3.1.3",

"resolved": "https://registry.npmjs.org/anymatch/-/anymatch-3.1.3.tgz",

"integrity": "sha512-KMReFUr0B4t+D+OBkjR3KYqvocp2XaSzO55UcB6mgQMd3KbcE+mWTyvVV7D/zsdEbNnV6acZUutkiHQXvTr1Rw==",

"dev": true,

"license": "ISC",

"dependencies": {

"normalize-path": "^3.0.0",

"picomatch": "^2.0.4"

},

"engines": {

"node": ">= 8"

}

},

"node\_modules/arg": {

"version": "5.0.2",

"resolved": "https://registry.npmjs.org/arg/-/arg-5.0.2.tgz",

"integrity": "sha512-PYjyFOLKQ9y57JvQ6QLo8dAgNqswh8M1RMJYdQduT6xbWSgK36P/Z/v+p888pM69jMMfS8Xd8F6I1kQ/I9HUGg==",

"dev": true,

"license": "MIT"

},

"node\_modules/argparse": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/argparse/-/argparse-2.0.1.tgz",

"integrity": "sha512-8+9WqebbFzpX9OR+Wa6O29asIogeRMzcGtAINdpMHHyAg10f05aSFVBbcEqGf/PXw1EjAZ+q2/bEBg3DvurK3Q==",

"dev": true,

"license": "Python-2.0"

},

"node\_modules/array-buffer-byte-length": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/array-buffer-byte-length/-/array-buffer-byte-length-1.0.1.tgz",

"integrity": "sha512-ahC5W1xgou+KTXix4sAO8Ki12Q+jf4i0+tmk3sC+zgcynshkHxzpXdImBehiUYKKKDwvfFiJl1tZt6ewscS1Mg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.5",

"is-array-buffer": "^3.0.4"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/array-includes": {

"version": "3.1.8",

"resolved": "https://registry.npmjs.org/array-includes/-/array-includes-3.1.8.tgz",

"integrity": "sha512-itaWrbYbqpGXkGhZPGUulwnhVf5Hpy1xiCFsGqyIGglbBxmG5vSjxQen3/WGOjPpNEv1RtBLKxbmVXm8HpJStQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.2",

"es-object-atoms": "^1.0.0",

"get-intrinsic": "^1.2.4",

"is-string": "^1.0.7"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/array.prototype.findlast": {

"version": "1.2.5",

"resolved": "https://registry.npmjs.org/array.prototype.findlast/-/array.prototype.findlast-1.2.5.tgz",

"integrity": "sha512-CVvd6FHg1Z3POpBLxO6E6zr+rSKEQ9L6rZHAaY7lLfhKsWYUBBOuMs0e9o24oopj6H+geRCX0YJ+TJLBK2eHyQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.2",

"es-errors": "^1.3.0",

"es-object-atoms": "^1.0.0",

"es-shim-unscopables": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/array.prototype.flat": {

"version": "1.3.2",

"resolved": "https://registry.npmjs.org/array.prototype.flat/-/array.prototype.flat-1.3.2.tgz",

"integrity": "sha512-djYB+Zx2vLewY8RWlNCUdHjDXs2XOgm602S9E7P/UpHgfeHL00cRiIF+IN/G/aUJ7kGPb6yO/ErDI5V2s8iycA==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"define-properties": "^1.2.0",

"es-abstract": "^1.22.1",

"es-shim-unscopables": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/array.prototype.flatmap": {

"version": "1.3.2",

"resolved": "https://registry.npmjs.org/array.prototype.flatmap/-/array.prototype.flatmap-1.3.2.tgz",

"integrity": "sha512-Ewyx0c9PmpcsByhSW4r+9zDU7sGjFc86qf/kKtuSCRdhfbk0SNLLkaT5qvcHnRGgc5NP/ly/y+qkXkqONX54CQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"define-properties": "^1.2.0",

"es-abstract": "^1.22.1",

"es-shim-unscopables": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/array.prototype.tosorted": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/array.prototype.tosorted/-/array.prototype.tosorted-1.1.4.tgz",

"integrity": "sha512-p6Fx8B7b7ZhL/gmUsAy0D15WhvDccw3mnGNbZpi3pmeJdxtWsj2jEaI4Y6oo3XiHfzuSgPwKc04MYt6KgvC/wA==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.3",

"es-errors": "^1.3.0",

"es-shim-unscopables": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/arraybuffer.prototype.slice": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/arraybuffer.prototype.slice/-/arraybuffer.prototype.slice-1.0.3.tgz",

"integrity": "sha512-bMxMKAjg13EBSVscxTaYA4mRc5t1UAXa2kXiGTNfZ079HIWXEkKmkgFrh/nJqamaLSrXO5H4WFFkPEaLJWbs3A==",

"dev": true,

"license": "MIT",

"dependencies": {

"array-buffer-byte-length": "^1.0.1",

"call-bind": "^1.0.5",

"define-properties": "^1.2.1",

"es-abstract": "^1.22.3",

"es-errors": "^1.2.1",

"get-intrinsic": "^1.2.3",

"is-array-buffer": "^3.0.4",

"is-shared-array-buffer": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/asynckit": {

"version": "0.4.0",

"resolved": "https://registry.npmjs.org/asynckit/-/asynckit-0.4.0.tgz",

"integrity": "sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0Q==",

"license": "MIT"

},

"node\_modules/autoprefixer": {

"version": "10.4.20",

"resolved": "https://registry.npmjs.org/autoprefixer/-/autoprefixer-10.4.20.tgz",

"integrity": "sha512-XY25y5xSv/wEoqzDyXXME4AFfkZI0P23z6Fs3YgymDnKJkCGOnkL0iTxCa85UTqaSgfcqyf3UA6+c7wUvx/16g==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/postcss/"

},

{

"type": "tidelift",

"url": "https://tidelift.com/funding/github/npm/autoprefixer"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"dependencies": {

"browserslist": "^4.23.3",

"caniuse-lite": "^1.0.30001646",

"fraction.js": "^4.3.7",

"normalize-range": "^0.1.2",

"picocolors": "^1.0.1",

"postcss-value-parser": "^4.2.0"

},

"bin": {

"autoprefixer": "bin/autoprefixer"

},

"engines": {

"node": "^10 || ^12 || >=14"

},

"peerDependencies": {

"postcss": "^8.1.0"

}

},

"node\_modules/available-typed-arrays": {

"version": "1.0.7",

"resolved": "https://registry.npmjs.org/available-typed-arrays/-/available-typed-arrays-1.0.7.tgz",

"integrity": "sha512-wvUjBtSGN7+7SjNpq/9M2Tg350UZD3q62IFZLbRAR1bSMlCo1ZaeW+BJ+D090e4hIIZLBcTDWe4Mh4jvUDajzQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"possible-typed-array-names": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/axios": {

"version": "1.7.3",

"resolved": "https://registry.npmjs.org/axios/-/axios-1.7.3.tgz",

"integrity": "sha512-Ar7ND9pU99eJ9GpoGQKhKf58GpUOgnzuaB7ueNQ5BMi0p+LZ5oaEnfF999fAArcTIBwXTCHAmGcHOZJaWPq9Nw==",

"license": "MIT",

"dependencies": {

"follow-redirects": "^1.15.6",

"form-data": "^4.0.0",

"proxy-from-env": "^1.1.0"

}

},

"node\_modules/balanced-match": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",

"integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==",

"dev": true,

"license": "MIT"

},

"node\_modules/binary-extensions": {

"version": "2.3.0",

"resolved": "https://registry.npmjs.org/binary-extensions/-/binary-extensions-2.3.0.tgz",

"integrity": "sha512-Ceh+7ox5qe7LJuLHoY0feh3pHuUDHAcRUeyL2VYghZwfpkNIy/+8Ocg0a3UuSoYzavmylwuLWQOf3hl0jjMMIw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/brace-expansion": {

"version": "1.1.11",

"resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.11.tgz",

"integrity": "sha512-iCuPHDFgrHX7H2vEI/5xpz07zSHB00TpugqhmYtVmMO6518mCuRMoOYFldEBl0g187ufozdaHgWKcYFb61qGiA==",

"dev": true,

"license": "MIT",

"dependencies": {

"balanced-match": "^1.0.0",

"concat-map": "0.0.1"

}

},

"node\_modules/braces": {

"version": "3.0.3",

"resolved": "https://registry.npmjs.org/braces/-/braces-3.0.3.tgz",

"integrity": "sha512-yQbXgO/OSZVD2IsiLlro+7Hf6Q18EJrKSEsdoMzKePKXct3gvD8oLcOQdIzGupr5Fj+EDe8gO/lxc1BzfMpxvA==",

"dev": true,

"license": "MIT",

"dependencies": {

"fill-range": "^7.1.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/browserslist": {

"version": "4.23.3",

"resolved": "https://registry.npmjs.org/browserslist/-/browserslist-4.23.3.tgz",

"integrity": "sha512-btwCFJVjI4YWDNfau8RhZ+B1Q/VLoUITrm3RlP6y1tYGWIOa+InuYiRGXUBXo8nA1qKmHMyLB/iVQg5TT4eFoA==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/browserslist"

},

{

"type": "tidelift",

"url": "https://tidelift.com/funding/github/npm/browserslist"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"dependencies": {

"caniuse-lite": "^1.0.30001646",

"electron-to-chromium": "^1.5.4",

"node-releases": "^2.0.18",

"update-browserslist-db": "^1.1.0"

},

"bin": {

"browserslist": "cli.js"

},

"engines": {

"node": "^6 || ^7 || ^8 || ^9 || ^10 || ^11 || ^12 || >=13.7"

}

},

"node\_modules/call-bind": {

"version": "1.0.7",

"resolved": "https://registry.npmjs.org/call-bind/-/call-bind-1.0.7.tgz",

"integrity": "sha512-GHTSNSYICQ7scH7sZ+M2rFopRoLh8t2bLSW6BbgrtLsahOIB5iyAVJf9GjWK3cYTDaMj4XdBpM1cA6pIS0Kv2w==",

"dev": true,

"license": "MIT",

"dependencies": {

"es-define-property": "^1.0.0",

"es-errors": "^1.3.0",

"function-bind": "^1.1.2",

"get-intrinsic": "^1.2.4",

"set-function-length": "^1.2.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/callsites": {

"version": "3.1.0",

"resolved": "https://registry.npmjs.org/callsites/-/callsites-3.1.0.tgz",

"integrity": "sha512-P8BjAsXvZS+VIDUI11hHCQEv74YT67YUi5JJFNWIqL235sBmjX4+qx9Muvls5ivyNENctx46xQLQ3aTuE7ssaQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6"

}

},

"node\_modules/camelcase-css": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/camelcase-css/-/camelcase-css-2.0.1.tgz",

"integrity": "sha512-QOSvevhslijgYwRx6Rv7zKdMF8lbRmx+uQGx2+vDc+KI/eBnsy9kit5aj23AgGu3pa4t9AgwbnXWqS+iOY+2aA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 6"

}

},

"node\_modules/caniuse-lite": {

"version": "1.0.30001651",

"resolved": "https://registry.npmjs.org/caniuse-lite/-/caniuse-lite-1.0.30001651.tgz",

"integrity": "sha512-9Cf+Xv1jJNe1xPZLGuUXLNkE1BoDkqRqYyFJ9TDYSqhduqA4hu4oR9HluGoWYQC/aj8WHjsGVV+bwkh0+tegRg==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/browserslist"

},

{

"type": "tidelift",

"url": "https://tidelift.com/funding/github/npm/caniuse-lite"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "CC-BY-4.0"

},

"node\_modules/chalk": {

"version": "2.4.2",

"resolved": "https://registry.npmjs.org/chalk/-/chalk-2.4.2.tgz",

"integrity": "sha512-Mti+f9lpJNcwF4tWV8/OrTTtF1gZi+f8FqlyAdouralcFWFQWF2+NgCHShjkCb+IFBLq9buZwE1xckQU4peSuQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-styles": "^3.2.1",

"escape-string-regexp": "^1.0.5",

"supports-color": "^5.3.0"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/chokidar": {

"version": "3.6.0",

"resolved": "https://registry.npmjs.org/chokidar/-/chokidar-3.6.0.tgz",

"integrity": "sha512-7VT13fmjotKpGipCW9JEQAusEPE+Ei8nl6/g4FBAmIm0GOOLMua9NDDo/DWp0ZAxCr3cPq5ZpBqmPAQgDda2Pw==",

"dev": true,

"license": "MIT",

"dependencies": {

"anymatch": "~3.1.2",

"braces": "~3.0.2",

"glob-parent": "~5.1.2",

"is-binary-path": "~2.1.0",

"is-glob": "~4.0.1",

"normalize-path": "~3.0.0",

"readdirp": "~3.6.0"

},

"engines": {

"node": ">= 8.10.0"

},

"funding": {

"url": "https://paulmillr.com/funding/"

},

"optionalDependencies": {

"fsevents": "~2.3.2"

}

},

"node\_modules/chokidar/node\_modules/glob-parent": {

"version": "5.1.2",

"resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-5.1.2.tgz",

"integrity": "sha512-AOIgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==",

"dev": true,

"license": "ISC",

"dependencies": {

"is-glob": "^4.0.1"

},

"engines": {

"node": ">= 6"

}

},

"node\_modules/clsx": {

"version": "2.1.1",

"resolved": "https://registry.npmjs.org/clsx/-/clsx-2.1.1.tgz",

"integrity": "sha512-eYm0QWBtUrBWZWG0d386OGAw16Z995PiOVo2B7bjWSbHedGl5e0ZWaq65kOGgUSNesEIDkB9ISbTg/JK9dhCZA==",

"license": "MIT",

"engines": {

"node": ">=6"

}

},

"node\_modules/color-convert": {

"version": "1.9.3",

"resolved": "https://registry.npmjs.org/color-convert/-/color-convert-1.9.3.tgz",

"integrity": "sha512-QfAUtd+vFdAtFQcC8CCyYt1fYWxSqAiK2cSD6zDB8N3cpsEBAvRxp9zOGg6G/SHHJYAT88/az/IuDGALsNVbGg==",

"dev": true,

"license": "MIT",

"dependencies": {

"color-name": "1.1.3"

}

},

"node\_modules/color-name": {

"version": "1.1.3",

"resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.3.tgz",

"integrity": "sha512-72fSenhMw2HZMTVHeCA9KCmpEIbzWiQsjN+BHcBbS9vr1mtt+vJjPdksIBNUmKAW8TFUDPJK5SUU3QhE9NEXDw==",

"dev": true,

"license": "MIT"

},

"node\_modules/combined-stream": {

"version": "1.0.8",

"resolved": "https://registry.npmjs.org/combined-stream/-/combined-stream-1.0.8.tgz",

"integrity": "sha512-FQN4MRfuJeHf7cBbBMJFXhKSDq+2kAArBlmRBvcvFE5BB1HZKXtSFASDhdlz9zOYwxh8lDdnvmMOe/+5cdoEdg==",

"license": "MIT",

"dependencies": {

"delayed-stream": "~1.0.0"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/commander": {

"version": "4.1.1",

"resolved": "https://registry.npmjs.org/commander/-/commander-4.1.1.tgz",

"integrity": "sha512-NOKm8xhkzAjzFx8B2v5OAHT+u5pRQc2UCa2Vq9jYL/31o2wi9mxBA7LIFs3sV5VSC49z6pEhfbMULvShKj26WA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 6"

}

},

"node\_modules/concat-map": {

"version": "0.0.1",

"resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",

"integrity": "sha512-/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkhKmqg==",

"dev": true,

"license": "MIT"

},

"node\_modules/convert-source-map": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/convert-source-map/-/convert-source-map-2.0.0.tgz",

"integrity": "sha512-Kvp459HrV2FEJ1CAsi1Ku+MY3kasH19TFykTz2xWmMeq6bk2NU3XXvfJ+Q61m0xktWwt+1HSYf3JZsTms3aRJg==",

"dev": true,

"license": "MIT"

},

"node\_modules/cross-spawn": {

"version": "7.0.3",

"resolved": "https://registry.npmjs.org/cross-spawn/-/cross-spawn-7.0.3.tgz",

"integrity": "sha512-iRDPJKUPVEND7dHPO8rkbOnPpyDygcDFtWjpeWNCgy8WP2rXcxXL8TskReQl6OrB2G7+UJrags1q15Fudc7G6w==",

"dev": true,

"license": "MIT",

"dependencies": {

"path-key": "^3.1.0",

"shebang-command": "^2.0.0",

"which": "^2.0.1"

},

"engines": {

"node": ">= 8"

}

},

"node\_modules/cssesc": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/cssesc/-/cssesc-3.0.0.tgz",

"integrity": "sha512-/Tb/JcjK111nNScGob5MNtsntNM1aCNUDipB/TkwZFhyDrrE47SOx/18wF2bbjgc3ZzCSKW1T5nt5EbFoAz/Vg==",

"dev": true,

"license": "MIT",

"bin": {

"cssesc": "bin/cssesc"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/csstype": {

"version": "3.1.3",

"resolved": "https://registry.npmjs.org/csstype/-/csstype-3.1.3.tgz",

"integrity": "sha512-M1uQkMl8rQK/szD0LNhtqxIPLpimGm8sOBwU7lLnCpSbTyY3yeU1Vc7l4KT5zT4s/yOxHH5O7tIuuLOCnLADRw==",

"dev": true,

"license": "MIT"

},

"node\_modules/data-view-buffer": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/data-view-buffer/-/data-view-buffer-1.0.1.tgz",

"integrity": "sha512-0lht7OugA5x3iJLOWFhWK/5ehONdprk0ISXqVFn/NFrDu+cuc8iADFrGQz5BnRK7LLU3JmkbXSxaqX+/mXYtUA==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.6",

"es-errors": "^1.3.0",

"is-data-view": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/data-view-byte-length": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/data-view-byte-length/-/data-view-byte-length-1.0.1.tgz",

"integrity": "sha512-4J7wRJD3ABAzr8wP+OcIcqq2dlUKp4DVflx++hs5h5ZKydWMI6/D/fAot+yh6g2tHh8fLFTvNOaVN357NvSrOQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"es-errors": "^1.3.0",

"is-data-view": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/data-view-byte-offset": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/data-view-byte-offset/-/data-view-byte-offset-1.0.0.tgz",

"integrity": "sha512-t/Ygsytq+R995EJ5PZlD4Cu56sWa8InXySaViRzw9apusqsOO2bQP+SbYzAhR0pFKoB+43lYy8rWban9JSuXnA==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.6",

"es-errors": "^1.3.0",

"is-data-view": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/debug": {

"version": "4.3.6",

"resolved": "https://registry.npmjs.org/debug/-/debug-4.3.6.tgz",

"integrity": "sha512-O/09Bd4Z1fBrU4VzkhFqVgpPzaGbw6Sm9FEkBT1A/YBXQFGuuSxa1dN2nxgxS34JmKXqYx8CZAwEVoJFImUXIg==",

"dev": true,

"license": "MIT",

"dependencies": {

"ms": "2.1.2"

},

"engines": {

"node": ">=6.0"

},

"peerDependenciesMeta": {

"supports-color": {

"optional": true

}

}

},

"node\_modules/deep-is": {

"version": "0.1.4",

"resolved": "https://registry.npmjs.org/deep-is/-/deep-is-0.1.4.tgz",

"integrity": "sha512-oIPzksmTg4/MriiaYGO+okXDT7ztn/w3Eptv/+gSIdMdKsJo0u4CfYNFJPy+4SKMuCqGw2wxnA+URMg3t8a/bQ==",

"dev": true,

"license": "MIT"

},

"node\_modules/define-data-property": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/define-data-property/-/define-data-property-1.1.4.tgz",

"integrity": "sha512-rBMvIzlpA8v6E+SJZoo++HAYqsLrkg7MSfIinMPFhmkorw7X+dOXVJQs+QT69zGkzMyfDnIMN2Wid1+NbL3T+A==",

"dev": true,

"license": "MIT",

"dependencies": {

"es-define-property": "^1.0.0",

"es-errors": "^1.3.0",

"gopd": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/define-properties": {

"version": "1.2.1",

"resolved": "https://registry.npmjs.org/define-properties/-/define-properties-1.2.1.tgz",

"integrity": "sha512-8QmQKqEASLd5nx0U1B1okLElbUuuttJ/AnYmRXbbbGDWh6uS208EjD4Xqq/I9wK7u0v6O08XhTWnt5XtEbR6Dg==",

"dev": true,

"license": "MIT",

"dependencies": {

"define-data-property": "^1.0.1",

"has-property-descriptors": "^1.0.0",

"object-keys": "^1.1.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/delayed-stream": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/delayed-stream/-/delayed-stream-1.0.0.tgz",

"integrity": "sha512-ZySD7Nf91aLB0RxL4KGrKHBXl7Eds1DAmEdcoVawXnLD7SDhpNgtuII2aAkg7a7QS41jxPSZ17p4VdGnMHk3MQ==",

"license": "MIT",

"engines": {

"node": ">=0.4.0"

}

},

"node\_modules/didyoumean": {

"version": "1.2.2",

"resolved": "https://registry.npmjs.org/didyoumean/-/didyoumean-1.2.2.tgz",

"integrity": "sha512-gxtyfqMg7GKyhQmb056K7M3xszy/myH8w+B4RT+QXBQsvAOdc3XymqDDPHx1BgPgsdAA5SIifona89YtRATDzw==",

"dev": true,

"license": "Apache-2.0"

},

"node\_modules/dlv": {

"version": "1.1.3",

"resolved": "https://registry.npmjs.org/dlv/-/dlv-1.1.3.tgz",

"integrity": "sha512-+HlytyjlPKnIG8XuRG8WvmBP8xs8P71y+SKKS6ZXWoEgLuePxtDoUEiH7WkdePWrQ5JBpE6aoVqfZfJUQkjXwA==",

"dev": true,

"license": "MIT"

},

"node\_modules/doctrine": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/doctrine/-/doctrine-3.0.0.tgz",

"integrity": "sha512-yS+Q5i3hBf7GBkd4KG8a7eBNNWNGLTaEwwYWUijIYM7zrlYDM0BFXHjjPWlWZ1Rg7UaddZeIDmi9jF3HmqiQ2w==",

"dev": true,

"license": "Apache-2.0",

"dependencies": {

"esutils": "^2.0.2"

},

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/eastasianwidth": {

"version": "0.2.0",

"resolved": "https://registry.npmjs.org/eastasianwidth/-/eastasianwidth-0.2.0.tgz",

"integrity": "sha512-I88TYZWc9XiYHRQ4/3c5rjjfgkjhLyW2luGIheGERbNQ6OY7yTybanSpDXZa8y7VUP9YmDcYa+eyq4ca7iLqWA==",

"dev": true,

"license": "MIT"

},

"node\_modules/electron-to-chromium": {

"version": "1.5.6",

"resolved": "https://registry.npmjs.org/electron-to-chromium/-/electron-to-chromium-1.5.6.tgz",

"integrity": "sha512-jwXWsM5RPf6j9dPYzaorcBSUg6AiqocPEyMpkchkvntaH9HGfOOMZwxMJjDY/XEs3T5dM7uyH1VhRMkqUU9qVw==",

"dev": true,

"license": "ISC"

},

"node\_modules/emoji-regex": {

"version": "9.2.2",

"resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-9.2.2.tgz",

"integrity": "sha512-L18DaJsXSUk2+42pv8mLs5jJT2hqFkFE4j21wOmgbUqsZ2hL72NsUU785g9RXgo3s0ZNgVl42TiHp3ZtOv/Vyg==",

"dev": true,

"license": "MIT"

},

"node\_modules/es-abstract": {

"version": "1.23.3",

"resolved": "https://registry.npmjs.org/es-abstract/-/es-abstract-1.23.3.tgz",

"integrity": "sha512-e+HfNH61Bj1X9/jLc5v1owaLYuHdeHHSQlkhCBiTK8rBvKaULl/beGMxwrMXjpYrv4pz22BlY570vVePA2ho4A==",

"dev": true,

"license": "MIT",

"dependencies": {

"array-buffer-byte-length": "^1.0.1",

"arraybuffer.prototype.slice": "^1.0.3",

"available-typed-arrays": "^1.0.7",

"call-bind": "^1.0.7",

"data-view-buffer": "^1.0.1",

"data-view-byte-length": "^1.0.1",

"data-view-byte-offset": "^1.0.0",

"es-define-property": "^1.0.0",

"es-errors": "^1.3.0",

"es-object-atoms": "^1.0.0",

"es-set-tostringtag": "^2.0.3",

"es-to-primitive": "^1.2.1",

"function.prototype.name": "^1.1.6",

"get-intrinsic": "^1.2.4",

"get-symbol-description": "^1.0.2",

"globalthis": "^1.0.3",

"gopd": "^1.0.1",

"has-property-descriptors": "^1.0.2",

"has-proto": "^1.0.3",

"has-symbols": "^1.0.3",

"hasown": "^2.0.2",

"internal-slot": "^1.0.7",

"is-array-buffer": "^3.0.4",

"is-callable": "^1.2.7",

"is-data-view": "^1.0.1",

"is-negative-zero": "^2.0.3",

"is-regex": "^1.1.4",

"is-shared-array-buffer": "^1.0.3",

"is-string": "^1.0.7",

"is-typed-array": "^1.1.13",

"is-weakref": "^1.0.2",

"object-inspect": "^1.13.1",

"object-keys": "^1.1.1",

"object.assign": "^4.1.5",

"regexp.prototype.flags": "^1.5.2",

"safe-array-concat": "^1.1.2",

"safe-regex-test": "^1.0.3",

"string.prototype.trim": "^1.2.9",

"string.prototype.trimend": "^1.0.8",

"string.prototype.trimstart": "^1.0.8",

"typed-array-buffer": "^1.0.2",

"typed-array-byte-length": "^1.0.1",

"typed-array-byte-offset": "^1.0.2",

"typed-array-length": "^1.0.6",

"unbox-primitive": "^1.0.2",

"which-typed-array": "^1.1.15"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/es-define-property": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.0.tgz",

"integrity": "sha512-jxayLKShrEqqzJ0eumQbVhTYQM27CfT1T35+gCgDFoL82JLsXqTJ76zv6A0YLOgEnLUMvLzsDsGIrl8NFpT2gQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"get-intrinsic": "^1.2.4"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-errors": {

"version": "1.3.0",

"resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",

"integrity": "sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzIlypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-iterator-helpers": {

"version": "1.0.19",

"resolved": "https://registry.npmjs.org/es-iterator-helpers/-/es-iterator-helpers-1.0.19.tgz",

"integrity": "sha512-zoMwbCcH5hwUkKJkT8kDIBZSz9I6mVG//+lDCinLCGov4+r7NIy0ld8o03M0cJxl2spVf6ESYVS6/gpIfq1FFw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.3",

"es-errors": "^1.3.0",

"es-set-tostringtag": "^2.0.3",

"function-bind": "^1.1.2",

"get-intrinsic": "^1.2.4",

"globalthis": "^1.0.3",

"has-property-descriptors": "^1.0.2",

"has-proto": "^1.0.3",

"has-symbols": "^1.0.3",

"internal-slot": "^1.0.7",

"iterator.prototype": "^1.1.2",

"safe-array-concat": "^1.1.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-object-atoms": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.0.0.tgz",

"integrity": "sha512-MZ4iQ6JwHOBQjahnjwaC1ZtIBH+2ohjamzAO3oaHcXYup7qxjF2fixyH+Q71voWHeOkI2q/TnJao/KfXYIZWbw==",

"dev": true,

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-set-tostringtag": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/es-set-tostringtag/-/es-set-tostringtag-2.0.3.tgz",

"integrity": "sha512-3T8uNMC3OQTHkFUsFq8r/BwAXLHvU/9O9mE0fBc/MY5iq/8H7ncvO947LmYA6ldWw9Uh8Yhf25zu6n7nML5QWQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"get-intrinsic": "^1.2.4",

"has-tostringtag": "^1.0.2",

"hasown": "^2.0.1"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-shim-unscopables": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/es-shim-unscopables/-/es-shim-unscopables-1.0.2.tgz",

"integrity": "sha512-J3yBRXCzDu4ULnQwxyToo/OjdMx6akgVC7K6few0a7F/0wLtmKKN7I73AH5T2836UuXRqN7Qg+IIUw/+YJksRw==",

"dev": true,

"license": "MIT",

"dependencies": {

"hasown": "^2.0.0"

}

},

"node\_modules/es-to-primitive": {

"version": "1.2.1",

"resolved": "https://registry.npmjs.org/es-to-primitive/-/es-to-primitive-1.2.1.tgz",

"integrity": "sha512-QCOllgZJtaUo9miYBcLChTUaHNjJF3PYs1VidD7AwiEj1kYxKeQTctLAezAOH5ZKRH0g2IgPn6KwB4IT8iRpvA==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-callable": "^1.1.4",

"is-date-object": "^1.0.1",

"is-symbol": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/esbuild": {

"version": "0.21.5",

"resolved": "https://registry.npmjs.org/esbuild/-/esbuild-0.21.5.tgz",

"integrity": "sha512-mg3OPMV4hXywwpoDxu3Qda5xCKQi+vCTZq8S9J/EpkhB2HzKXq4SNFZE3+NK93JYxc8VMSep+lOUSC/RVKaBqw==",

"dev": true,

"hasInstallScript": true,

"license": "MIT",

"bin": {

"esbuild": "bin/esbuild"

},

"engines": {

"node": ">=12"

},

"optionalDependencies": {

"@esbuild/aix-ppc64": "0.21.5",

"@esbuild/android-arm": "0.21.5",

"@esbuild/android-arm64": "0.21.5",

"@esbuild/android-x64": "0.21.5",

"@esbuild/darwin-arm64": "0.21.5",

"@esbuild/darwin-x64": "0.21.5",

"@esbuild/freebsd-arm64": "0.21.5",

"@esbuild/freebsd-x64": "0.21.5",

"@esbuild/linux-arm": "0.21.5",

"@esbuild/linux-arm64": "0.21.5",

"@esbuild/linux-ia32": "0.21.5",

"@esbuild/linux-loong64": "0.21.5",

"@esbuild/linux-mips64el": "0.21.5",

"@esbuild/linux-ppc64": "0.21.5",

"@esbuild/linux-riscv64": "0.21.5",

"@esbuild/linux-s390x": "0.21.5",

"@esbuild/linux-x64": "0.21.5",

"@esbuild/netbsd-x64": "0.21.5",

"@esbuild/openbsd-x64": "0.21.5",

"@esbuild/sunos-x64": "0.21.5",

"@esbuild/win32-arm64": "0.21.5",

"@esbuild/win32-ia32": "0.21.5",

"@esbuild/win32-x64": "0.21.5"

}

},

"node\_modules/escalade": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/escalade/-/escalade-3.1.2.tgz",

"integrity": "sha512-ErCHMCae19vR8vQGe50xIsVomy19rg6gFu3+r3jkEO46suLMWBksvVyoGgQV+jOfl84ZSOSlmv6Gxa89PmTGmA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6"

}

},

"node\_modules/escape-string-regexp": {

"version": "1.0.5",

"resolved": "https://registry.npmjs.org/escape-string-regexp/-/escape-string-regexp-1.0.5.tgz",

"integrity": "sha512-vbRorB5FUQWvla16U8R/qgaFIya2qGzwDrNmCZuYKrbdSUMG6I1ZCGQRefkRVhuOkIGVne7BQ35DSfo1qvJqFg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.8.0"

}

},

"node\_modules/eslint": {

"version": "8.57.0",

"resolved": "https://registry.npmjs.org/eslint/-/eslint-8.57.0.tgz",

"integrity": "sha512-dZ6+mexnaTIbSBZWgou51U6OmzIhYM2VcNdtiTtI7qPNZm35Akpr0f6vtw3w1Kmn5PYo+tZVfh13WrhpS6oLqQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"@eslint-community/eslint-utils": "^4.2.0",

"@eslint-community/regexpp": "^4.6.1",

"@eslint/eslintrc": "^2.1.4",

"@eslint/js": "8.57.0",

"@humanwhocodes/config-array": "^0.11.14",

"@humanwhocodes/module-importer": "^1.0.1",

"@nodelib/fs.walk": "^1.2.8",

"@ungap/structured-clone": "^1.2.0",

"ajv": "^6.12.4",

"chalk": "^4.0.0",

"cross-spawn": "^7.0.2",

"debug": "^4.3.2",

"doctrine": "^3.0.0",

"escape-string-regexp": "^4.0.0",

"eslint-scope": "^7.2.2",

"eslint-visitor-keys": "^3.4.3",

"espree": "^9.6.1",

"esquery": "^1.4.2",

"esutils": "^2.0.2",

"fast-deep-equal": "^3.1.3",

"file-entry-cache": "^6.0.1",

"find-up": "^5.0.0",

"glob-parent": "^6.0.2",

"globals": "^13.19.0",

"graphemer": "^1.4.0",

"ignore": "^5.2.0",

"imurmurhash": "^0.1.4",

"is-glob": "^4.0.0",

"is-path-inside": "^3.0.3",

"js-yaml": "^4.1.0",

"json-stable-stringify-without-jsonify": "^1.0.1",

"levn": "^0.4.1",

"lodash.merge": "^4.6.2",

"minimatch": "^3.1.2",

"natural-compare": "^1.4.0",

"optionator": "^0.9.3",

"strip-ansi": "^6.0.1",

"text-table": "^0.2.0"

},

"bin": {

"eslint": "bin/eslint.js"

},

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

},

"funding": {

"url": "https://opencollective.com/eslint"

}

},

"node\_modules/eslint-plugin-react": {

"version": "7.35.0",

"resolved": "https://registry.npmjs.org/eslint-plugin-react/-/eslint-plugin-react-7.35.0.tgz",

"integrity": "sha512-v501SSMOWv8gerHkk+IIQBkcGRGrO2nfybfj5pLxuJNFTPxxA3PSryhXTK+9pNbtkggheDdsC0E9Q8CuPk6JKA==",

"dev": true,

"license": "MIT",

"dependencies": {

"array-includes": "^3.1.8",

"array.prototype.findlast": "^1.2.5",

"array.prototype.flatmap": "^1.3.2",

"array.prototype.tosorted": "^1.1.4",

"doctrine": "^2.1.0",

"es-iterator-helpers": "^1.0.19",

"estraverse": "^5.3.0",

"hasown": "^2.0.2",

"jsx-ast-utils": "^2.4.1 || ^3.0.0",

"minimatch": "^3.1.2",

"object.entries": "^1.1.8",

"object.fromentries": "^2.0.8",

"object.values": "^1.2.0",

"prop-types": "^15.8.1",

"resolve": "^2.0.0-next.5",

"semver": "^6.3.1",

"string.prototype.matchall": "^4.0.11",

"string.prototype.repeat": "^1.0.0"

},

"engines": {

"node": ">=4"

},

"peerDependencies": {

"eslint": "^3 || ^4 || ^5 || ^6 || ^7 || ^8 || ^9.7"

}

},

"node\_modules/eslint-plugin-react-hooks": {

"version": "4.6.2",

"resolved": "https://registry.npmjs.org/eslint-plugin-react-hooks/-/eslint-plugin-react-hooks-4.6.2.tgz",

"integrity": "sha512-QzliNJq4GinDBcD8gPB5v0wh6g8q3SUi6EFF0x8N/BL9PoVs0atuGc47ozMRyOWAKdwaZ5OnbOEa3WR+dSGKuQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=10"

},

"peerDependencies": {

"eslint": "^3.0.0 || ^4.0.0 || ^5.0.0 || ^6.0.0 || ^7.0.0 || ^8.0.0-0"

}

},

"node\_modules/eslint-plugin-react-refresh": {

"version": "0.4.9",

"resolved": "https://registry.npmjs.org/eslint-plugin-react-refresh/-/eslint-plugin-react-refresh-0.4.9.tgz",

"integrity": "sha512-QK49YrBAo5CLNLseZ7sZgvgTy21E6NEw22eZqc4teZfH8pxV3yXc9XXOYfUI6JNpw7mfHNkAeWtBxrTyykB6HA==",

"dev": true,

"license": "MIT",

"peerDependencies": {

"eslint": ">=7"

}

},

"node\_modules/eslint-plugin-react/node\_modules/doctrine": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/doctrine/-/doctrine-2.1.0.tgz",

"integrity": "sha512-35mSku4ZXK0vfCuHEDAwt55dg2jNajHZ1odvF+8SSr82EsZY4QmXfuWso8oEd8zRhVObSN18aM0CjSdoBX7zIw==",

"dev": true,

"license": "Apache-2.0",

"dependencies": {

"esutils": "^2.0.2"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/eslint-scope": {

"version": "7.2.2",

"resolved": "https://registry.npmjs.org/eslint-scope/-/eslint-scope-7.2.2.tgz",

"integrity": "sha512-dOt21O7lTMhDM+X9mB4GX+DZrZtCUJPL/wlcTqxyrx5IvO0IYtILdtrQGQp+8n5S0gwSVmOf9NQrjMOgfQZlIg==",

"dev": true,

"license": "BSD-2-Clause",

"dependencies": {

"esrecurse": "^4.3.0",

"estraverse": "^5.2.0"

},

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

},

"funding": {

"url": "https://opencollective.com/eslint"

}

},

"node\_modules/eslint-visitor-keys": {

"version": "3.4.3",

"resolved": "https://registry.npmjs.org/eslint-visitor-keys/-/eslint-visitor-keys-3.4.3.tgz",

"integrity": "sha512-wpc+LXeiyiisxPlEkUzU6svyS1frIO3Mgxj1fdy7Pm8Ygzguax2N3Fa/D/ag1WqbOprdI+uY6wMUl8/a2G+iag==",

"dev": true,

"license": "Apache-2.0",

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

},

"funding": {

"url": "https://opencollective.com/eslint"

}

},

"node\_modules/eslint/node\_modules/ansi-styles": {

"version": "4.3.0",

"resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",

"integrity": "sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPlFp222wVALIheZJQSEg==",

"dev": true,

"license": "MIT",

"dependencies": {

"color-convert": "^2.0.1"

},

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/chalk/ansi-styles?sponsor=1"

}

},

"node\_modules/eslint/node\_modules/chalk": {

"version": "4.1.2",

"resolved": "https://registry.npmjs.org/chalk/-/chalk-4.1.2.tgz",

"integrity": "sha512-oKnbhFyRIXpUuez8iBMmyEa4nbj4IOQyuhc/wy9kY7/WVPcwIO9VA668Pu8RkO7+0G76SLROeyw9CpQ061i4mA==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-styles": "^4.1.0",

"supports-color": "^7.1.0"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/chalk/chalk?sponsor=1"

}

},

"node\_modules/eslint/node\_modules/color-convert": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",

"integrity": "sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHCaElKQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"color-name": "~1.1.4"

},

"engines": {

"node": ">=7.0.0"

}

},

"node\_modules/eslint/node\_modules/color-name": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",

"integrity": "sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGXgAsQA==",

"dev": true,

"license": "MIT"

},

"node\_modules/eslint/node\_modules/escape-string-regexp": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/escape-string-regexp/-/escape-string-regexp-4.0.0.tgz",

"integrity": "sha512-TtpcNJ3XAzx3Gq8sWRzJaVajRs0uVxA2YAkdb1jm2YkPz4G6egUFAyA3n5vtEIZefPk5Wa4UXbKuS5fKkJWdgA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/eslint/node\_modules/globals": {

"version": "13.24.0",

"resolved": "https://registry.npmjs.org/globals/-/globals-13.24.0.tgz",

"integrity": "sha512-AhO5QUcj8llrbG09iWhPU2B204J1xnPeL8kQmVorSsy+Sjj1sk8gIyh6cUocGmH4L0UuhAJy+hJMRA4mgA4mFQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"type-fest": "^0.20.2"

},

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/eslint/node\_modules/has-flag": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/has-flag/-/has-flag-4.0.0.tgz",

"integrity": "sha512-EykJT/Q1KjTWctppgIAgfSO0tKVuZUjhgMr17kqTumMl6Afv3EISleU7qZUzoXDFTAHTDC4NOoG/ZxU3EvlMPQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/eslint/node\_modules/supports-color": {

"version": "7.2.0",

"resolved": "https://registry.npmjs.org/supports-color/-/supports-color-7.2.0.tgz",

"integrity": "sha512-qpCAvRl9stuOHveKsn7HncJRvv501qIacKzQlO/+Lwxc9+0q2wLyv4Dfvt80/DPn2pqOBsJdDiogXGR9+OvwRw==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-flag": "^4.0.0"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/espree": {

"version": "9.6.1",

"resolved": "https://registry.npmjs.org/espree/-/espree-9.6.1.tgz",

"integrity": "sha512-oruZaFkjorTpF32kDSI5/75ViwGeZginGGy2NoOSg3Q9bnwlnmDm4HLnkl0RE3n+njDXR037aY1+x58Z/zFdwQ==",

"dev": true,

"license": "BSD-2-Clause",

"dependencies": {

"acorn": "^8.9.0",

"acorn-jsx": "^5.3.2",

"eslint-visitor-keys": "^3.4.1"

},

"engines": {

"node": "^12.22.0 || ^14.17.0 || >=16.0.0"

},

"funding": {

"url": "https://opencollective.com/eslint"

}

},

"node\_modules/esquery": {

"version": "1.6.0",

"resolved": "https://registry.npmjs.org/esquery/-/esquery-1.6.0.tgz",

"integrity": "sha512-ca9pw9fomFcKPvFLXhBKUK90ZvGibiGOvRJNbjljY7s7uq/5YO4BOzcYtJqExdx99rF6aAcnRxHmcUHcz6sQsg==",

"dev": true,

"license": "BSD-3-Clause",

"dependencies": {

"estraverse": "^5.1.0"

},

"engines": {

"node": ">=0.10"

}

},

"node\_modules/esrecurse": {

"version": "4.3.0",

"resolved": "https://registry.npmjs.org/esrecurse/-/esrecurse-4.3.0.tgz",

"integrity": "sha512-KmfKL3b6G+RXvP8N1vr3Tq1kL/oCFgn2NYXEtqP8/L3pKapUA4G8cFVaoF3SU323CD4XypR/ffioHmkti6/Tag==",

"dev": true,

"license": "BSD-2-Clause",

"dependencies": {

"estraverse": "^5.2.0"

},

"engines": {

"node": ">=4.0"

}

},

"node\_modules/estraverse": {

"version": "5.3.0",

"resolved": "https://registry.npmjs.org/estraverse/-/estraverse-5.3.0.tgz",

"integrity": "sha512-MMdARuVEQziNTeJD8DgMqmhwR11BRQ/cBP+pLtYdSTnf3MIO8fFeiINEbX36ZdNlfU/7A9f3gUw49B3oQsvwBA==",

"dev": true,

"license": "BSD-2-Clause",

"engines": {

"node": ">=4.0"

}

},

"node\_modules/esutils": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/esutils/-/esutils-2.0.3.tgz",

"integrity": "sha512-kVscqXk4OCp68SZ0dkgEKVi6/8ij300KBWTJq32P/dYeWTSwK41WyTxalN1eRmA5Z9UU/LX9D7FWSmV9SAYx6g==",

"dev": true,

"license": "BSD-2-Clause",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/fast-deep-equal": {

"version": "3.1.3",

"resolved": "https://registry.npmjs.org/fast-deep-equal/-/fast-deep-equal-3.1.3.tgz",

"integrity": "sha512-f3qQ9oQy9j2AhBe/H9VC91wLmKBCCU/gDOnKNAYG5hswO7BLKj09Hc5HYNz9cGI++xlpDCIgDaitVs03ATR84Q==",

"dev": true,

"license": "MIT"

},

"node\_modules/fast-glob": {

"version": "3.3.2",

"resolved": "https://registry.npmjs.org/fast-glob/-/fast-glob-3.3.2.tgz",

"integrity": "sha512-oX2ruAFQwf/Orj8m737Y5adxDQO0LAB7/S5MnxCdTNDd4p6BsyIVsv9JQsATbTSq8KHRpLwIHbVlUNatxd+1Ow==",

"dev": true,

"license": "MIT",

"dependencies": {

"@nodelib/fs.stat": "^2.0.2",

"@nodelib/fs.walk": "^1.2.3",

"glob-parent": "^5.1.2",

"merge2": "^1.3.0",

"micromatch": "^4.0.4"

},

"engines": {

"node": ">=8.6.0"

}

},

"node\_modules/fast-glob/node\_modules/glob-parent": {

"version": "5.1.2",

"resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-5.1.2.tgz",

"integrity": "sha512-AOIgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==",

"dev": true,

"license": "ISC",

"dependencies": {

"is-glob": "^4.0.1"

},

"engines": {

"node": ">= 6"

}

},

"node\_modules/fast-json-stable-stringify": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/fast-json-stable-stringify/-/fast-json-stable-stringify-2.1.0.tgz",

"integrity": "sha512-lhd/wF+Lk98HZoTCtlVraHtfh5XYijIjalXck7saUtuanSDyLMxnHhSXEDJqHxD7msR8D0uCmqlkwjCV8xvwHw==",

"dev": true,

"license": "MIT"

},

"node\_modules/fast-levenshtein": {

"version": "2.0.6",

"resolved": "https://registry.npmjs.org/fast-levenshtein/-/fast-levenshtein-2.0.6.tgz",

"integrity": "sha512-DCXu6Ifhqcks7TZKY3Hxp3y6qphY5SJZmrWMDrKcERSOXWQdMhU9Ig/PYrzyw/ul9jOIyh0N4M0tbC5hodg8dw==",

"dev": true,

"license": "MIT"

},

"node\_modules/fastq": {

"version": "1.17.1",

"resolved": "https://registry.npmjs.org/fastq/-/fastq-1.17.1.tgz",

"integrity": "sha512-sRVD3lWVIXWg6By68ZN7vho9a1pQcN/WBFaAAsDDFzlJjvoGx0P8z7V1t72grFJfJhu3YPZBuu25f7Kaw2jN1w==",

"dev": true,

"license": "ISC",

"dependencies": {

"reusify": "^1.0.4"

}

},

"node\_modules/file-entry-cache": {

"version": "6.0.1",

"resolved": "https://registry.npmjs.org/file-entry-cache/-/file-entry-cache-6.0.1.tgz",

"integrity": "sha512-7Gps/XWymbLk2QLYK4NzpMOrYjMhdIxXuIvy2QBsLE6ljuodKvdkWs/cpyJJ3CVIVpH0Oi1Hvg1ovbMzLdFBBg==",

"dev": true,

"license": "MIT",

"dependencies": {

"flat-cache": "^3.0.4"

},

"engines": {

"node": "^10.12.0 || >=12.0.0"

}

},

"node\_modules/fill-range": {

"version": "7.1.1",

"resolved": "https://registry.npmjs.org/fill-range/-/fill-range-7.1.1.tgz",

"integrity": "sha512-YsGpe3WHLK8ZYi4tWDg2Jy3ebRz2rXowDxnld4bkQB00cc/1Zw9AWnC0i9ztDJitivtQvaI9KaLyKrc+hBW0yg==",

"dev": true,

"license": "MIT",

"dependencies": {

"to-regex-range": "^5.0.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/find-up": {

"version": "5.0.0",

"resolved": "https://registry.npmjs.org/find-up/-/find-up-5.0.0.tgz",

"integrity": "sha512-78/PXT1wlLLDgTzDs7sjq9hzz0vXD+zn+7wypEe4fXQxCmdmqfGsEPQxmiCSQI3ajFV91bVSsvNtrJRiW6nGng==",

"dev": true,

"license": "MIT",

"dependencies": {

"locate-path": "^6.0.0",

"path-exists": "^4.0.0"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/flat-cache": {

"version": "3.2.0",

"resolved": "https://registry.npmjs.org/flat-cache/-/flat-cache-3.2.0.tgz",

"integrity": "sha512-CYcENa+FtcUKLmhhqyctpclsq7QF38pKjZHsGNiSQF5r4FtoKDWabFDl3hzaEQMvT1LHEysw5twgLvpYYb4vbw==",

"dev": true,

"license": "MIT",

"dependencies": {

"flatted": "^3.2.9",

"keyv": "^4.5.3",

"rimraf": "^3.0.2"

},

"engines": {

"node": "^10.12.0 || >=12.0.0"

}

},

"node\_modules/flatted": {

"version": "3.3.1",

"resolved": "https://registry.npmjs.org/flatted/-/flatted-3.3.1.tgz",

"integrity": "sha512-X8cqMLLie7KsNUDSdzeN8FYK9rEt4Dt67OsG/DNGnYTSDBG4uFAJFBnUeiV+zCVAvwFy56IjM9sH51jVaEhNxw==",

"dev": true,

"license": "ISC"

},

"node\_modules/follow-redirects": {

"version": "1.15.6",

"resolved": "https://registry.npmjs.org/follow-redirects/-/follow-redirects-1.15.6.tgz",

"integrity": "sha512-wWN62YITEaOpSK584EZXJafH1AGpO8RVgElfkuXbTOrPX4fIfOyEpW/CsiNd8JdYrAoOvafRTOEnvsO++qCqFA==",

"funding": [

{

"type": "individual",

"url": "https://github.com/sponsors/RubenVerborgh"

}

],

"license": "MIT",

"engines": {

"node": ">=4.0"

},

"peerDependenciesMeta": {

"debug": {

"optional": true

}

}

},

"node\_modules/for-each": {

"version": "0.3.3",

"resolved": "https://registry.npmjs.org/for-each/-/for-each-0.3.3.tgz",

"integrity": "sha512-jqYfLp7mo9vIyQf8ykW2v7A+2N4QjeCeI5+Dz9XraiO1ign81wjiH7Fb9vSOWvQfNtmSa4H2RoQTrrXivdUZmw==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-callable": "^1.1.3"

}

},

"node\_modules/foreground-child": {

"version": "3.3.0",

"resolved": "https://registry.npmjs.org/foreground-child/-/foreground-child-3.3.0.tgz",

"integrity": "sha512-Ld2g8rrAyMYFXBhEqMz8ZAHBi4J4uS1i/CxGMDnjyFWddMXLVcDp051DZfu+t7+ab7Wv6SMqpWmyFIj5UbfFvg==",

"dev": true,

"license": "ISC",

"dependencies": {

"cross-spawn": "^7.0.0",

"signal-exit": "^4.0.1"

},

"engines": {

"node": ">=14"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/form-data": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/form-data/-/form-data-4.0.0.tgz",

"integrity": "sha512-ETEklSGi5t0QMZuiXoA/Q6vcnxcLQP5vdugSpuAyi6SVGi2clPPp+xgEhuMaHC+zGgn31Kd235W35f7Hykkaww==",

"license": "MIT",

"dependencies": {

"asynckit": "^0.4.0",

"combined-stream": "^1.0.8",

"mime-types": "^2.1.12"

},

"engines": {

"node": ">= 6"

}

},

"node\_modules/fraction.js": {

"version": "4.3.7",

"resolved": "https://registry.npmjs.org/fraction.js/-/fraction.js-4.3.7.tgz",

"integrity": "sha512-ZsDfxO51wGAXREY55a7la9LScWpwv9RxIrYABrlvOFBlH/ShPnrtsXeuUIfXKKOVicNxQ+o8JTbJvjS4M89yew==",

"dev": true,

"license": "MIT",

"engines": {

"node": "\*"

},

"funding": {

"type": "patreon",

"url": "https://github.com/sponsors/rawify"

}

},

"node\_modules/fs.realpath": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/fs.realpath/-/fs.realpath-1.0.0.tgz",

"integrity": "sha512-OO0pH2lK6a0hZnAdau5ItzHPI6pUlvI7jMVnxUQRtw4owF2wk8lOSabtGDCTP4Ggrg2MbGnWO9X8K1t4+fGMDw==",

"dev": true,

"license": "ISC"

},

"node\_modules/fsevents": {

"version": "2.3.3",

"resolved": "https://registry.npmjs.org/fsevents/-/fsevents-2.3.3.tgz",

"integrity": "sha512-5xoDfX+fL7faATnagmWPpbFtwh/R77WmMMqqHGS65C3vvB0YHrgF+B1YmZ3441tMj5n63k0212XNoJwzlhffQw==",

"dev": true,

"hasInstallScript": true,

"license": "MIT",

"optional": true,

"os": [

"darwin"

],

"engines": {

"node": "^8.16.0 || ^10.6.0 || >=11.0.0"

}

},

"node\_modules/function-bind": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",

"integrity": "sha512-7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==",

"dev": true,

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/function.prototype.name": {

"version": "1.1.6",

"resolved": "https://registry.npmjs.org/function.prototype.name/-/function.prototype.name-1.1.6.tgz",

"integrity": "sha512-Z5kx79swU5P27WEayXM1tBi5Ze/lbIyiNgU3qyXUOf9b2rgXYyF9Dy9Cx+IQv/Lc8WCG6L82zwUPpSS9hGehIg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"define-properties": "^1.2.0",

"es-abstract": "^1.22.1",

"functions-have-names": "^1.2.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/functions-have-names": {

"version": "1.2.3",

"resolved": "https://registry.npmjs.org/functions-have-names/-/functions-have-names-1.2.3.tgz",

"integrity": "sha512-xckBUXyTIqT97tq2x2AMb+g163b5JFysYk0x4qxNFwbfQkmNZoiRHb6sPzI9/QV33WeuvVYBUIiD4NzNIyqaRQ==",

"dev": true,

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/gensync": {

"version": "1.0.0-beta.2",

"resolved": "https://registry.npmjs.org/gensync/-/gensync-1.0.0-beta.2.tgz",

"integrity": "sha512-3hN7NaskYvMDLQY55gnW3NQ+mesEAepTqlg+VEbj7zzqEMBVNhzcGYYeqFo/TlYz6eQiFcp1HcsCZO+nGgS8zg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6.9.0"

}

},

"node\_modules/get-intrinsic": {

"version": "1.2.4",

"resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.2.4.tgz",

"integrity": "sha512-5uYhsJH8VJBTv7oslg4BznJYhDoRI6waYCxMmCdnTrcCrHA/fCFKoTFz2JKKE0HdDFUF7/oQuhzumXJK7paBRQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"function-bind": "^1.1.2",

"has-proto": "^1.0.1",

"has-symbols": "^1.0.3",

"hasown": "^2.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/get-symbol-description": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/get-symbol-description/-/get-symbol-description-1.0.2.tgz",

"integrity": "sha512-g0QYk1dZBxGwk+Ngc+ltRH2IBp2f7zBkBMBJZCDerh6EhlhSR6+9irMCuT/09zD6qkarHUSn529sK/yL4S27mg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.5",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.4"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/glob": {

"version": "7.2.3",

"resolved": "https://registry.npmjs.org/glob/-/glob-7.2.3.tgz",

"integrity": "sha512-nFR0zLpU2YCaRxwoCJvL6UvCH2JFyFVIvwTLsIf21AuHlMskA1hhTdk+LlYJtOlYt9v6dvszD2BGRqBL+iQK9Q==",

"deprecated": "Glob versions prior to v9 are no longer supported",

"dev": true,

"license": "ISC",

"dependencies": {

"fs.realpath": "^1.0.0",

"inflight": "^1.0.4",

"inherits": "2",

"minimatch": "^3.1.1",

"once": "^1.3.0",

"path-is-absolute": "^1.0.0"

},

"engines": {

"node": "\*"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/glob-parent": {

"version": "6.0.2",

"resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-6.0.2.tgz",

"integrity": "sha512-XxwI8EOhVQgWp6iDL+3b0r86f4d6AX6zSU55HfB4ydCEuXLXc5FcYeOu+nnGftS4TEju/11rt4KJPTMgbfmv4A==",

"dev": true,

"license": "ISC",

"dependencies": {

"is-glob": "^4.0.3"

},

"engines": {

"node": ">=10.13.0"

}

},

"node\_modules/globals": {

"version": "11.12.0",

"resolved": "https://registry.npmjs.org/globals/-/globals-11.12.0.tgz",

"integrity": "sha512-WOBp/EEGUiIsJSp7wcv/y6MO+lV9UoncWqxuFfm8eBwzWNgyfBd6Gz+IeKQ9jCmyhoH99g15M3T+QaVHFjizVA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=4"

}

},

"node\_modules/globalthis": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/globalthis/-/globalthis-1.0.4.tgz",

"integrity": "sha512-DpLKbNU4WylpxJykQujfCcwYWiV/Jhm50Goo0wrVILAv5jOr9d+H+UR3PhSCD2rCCEIg0uc+G+muBTwD54JhDQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"define-properties": "^1.2.1",

"gopd": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/gopd": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/gopd/-/gopd-1.0.1.tgz",

"integrity": "sha512-d65bNlIadxvpb/A2abVdlqKqV563juRnZ1Wtk6s1sIR8uNsXR70xqIzVqxVf1eTqDunwT2MkczEeaezCKTZhwA==",

"dev": true,

"license": "MIT",

"dependencies": {

"get-intrinsic": "^1.1.3"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/graphemer": {

"version": "1.4.0",

"resolved": "https://registry.npmjs.org/graphemer/-/graphemer-1.4.0.tgz",

"integrity": "sha512-EtKwoO6kxCL9WO5xipiHTZlSzBm7WLT627TqC/uVRd0HKmq8NXyebnNYxDoBi7wt8eTWrUrKXCOVaFq9x1kgag==",

"dev": true,

"license": "MIT"

},

"node\_modules/has-bigints": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/has-bigints/-/has-bigints-1.0.2.tgz",

"integrity": "sha512-tSvCKtBr9lkF0Ex0aQiP9N+OpV4zi2r/Nee5VkRDbaqv35RLYMzbwQfFSZZH0kR+Rd6302UJZ2p/bJCEoR3VoQ==",

"dev": true,

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-flag": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/has-flag/-/has-flag-3.0.0.tgz",

"integrity": "sha512-sKJf1+ceQBr4SMkvQnBDNDtf4TXpVhVGateu0t918bl30FnbE2m4vNLX+VWe/dpjlb+HugGYzW7uQXH98HPEYw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=4"

}

},

"node\_modules/has-property-descriptors": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/has-property-descriptors/-/has-property-descriptors-1.0.2.tgz",

"integrity": "sha512-55JNKuIW+vq4Ke1BjOTjM2YctQIvCT7GFzHwmfZPGo5wnrgkid0YQtnAleFSqumZm4az3n2BS+erby5ipJdgrg==",

"dev": true,

"license": "MIT",

"dependencies": {

"es-define-property": "^1.0.0"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-proto": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/has-proto/-/has-proto-1.0.3.tgz",

"integrity": "sha512-SJ1amZAJUiZS+PhsVLf5tGydlaVB8EdFpaSO4gmiUKUOxk8qzn5AIy4ZeJUmh22znIdk/uMAUT2pl3FxzVUH+Q==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-symbols": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.0.3.tgz",

"integrity": "sha512-l3LCuF6MgDNwTDKkdYGEihYjt5pRPbEg46rtlmnSPlUbgmB8LOIrKJbYYFBSbnPaJexMKtiPO8hmeRjRz2Td+A==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-tostringtag": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/has-tostringtag/-/has-tostringtag-1.0.2.tgz",

"integrity": "sha512-NqADB8VjPFLM2V0VvHUewwwsw0ZWBaIdgo+ieHtK3hasLz4qeCRjYcqfB6AQrBggRKppKF8L52/VqdVsO47Dlw==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-symbols": "^1.0.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/hasown": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",

"integrity": "sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+CyS8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7bfgSjkUMQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"function-bind": "^1.1.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/ignore": {

"version": "5.3.1",

"resolved": "https://registry.npmjs.org/ignore/-/ignore-5.3.1.tgz",

"integrity": "sha512-5Fytz/IraMjqpwfd34ke28PTVMjZjJG2MPn5t7OE4eUCUNf8BAa7b5WUS9/Qvr6mwOQS7Mk6vdsMno5he+T8Xw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 4"

}

},

"node\_modules/import-fresh": {

"version": "3.3.0",

"resolved": "https://registry.npmjs.org/import-fresh/-/import-fresh-3.3.0.tgz",

"integrity": "sha512-veYYhQa+D1QBKznvhUHxb8faxlrwUnxseDAbAp457E0wLNio2bOSKnjYDhMj+YiAq61xrMGhQk9iXVk5FzgQMw==",

"dev": true,

"license": "MIT",

"dependencies": {

"parent-module": "^1.0.0",

"resolve-from": "^4.0.0"

},

"engines": {

"node": ">=6"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/imurmurhash": {

"version": "0.1.4",

"resolved": "https://registry.npmjs.org/imurmurhash/-/imurmurhash-0.1.4.tgz",

"integrity": "sha512-JmXMZ6wuvDmLiHEml9ykzqO6lwFbof0GG4IkcGaENdCRDDmMVnny7s5HsIgHCbaq0w2MyPhDqkhTUgS2LU2PHA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.8.19"

}

},

"node\_modules/inflight": {

"version": "1.0.6",

"resolved": "https://registry.npmjs.org/inflight/-/inflight-1.0.6.tgz",

"integrity": "sha512-k92I/b08q4wvFscXCLvqfsHCrjrF7yiXsQuIVvVE7N82W3+aqpzuUdBbfhWcy/FZR3/4IgflMgKLOsvPDrGCJA==",

"deprecated": "This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.",

"dev": true,

"license": "ISC",

"dependencies": {

"once": "^1.3.0",

"wrappy": "1"

}

},

"node\_modules/inherits": {

"version": "2.0.4",

"resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",

"integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4dYeZIQqewQ==",

"dev": true,

"license": "ISC"

},

"node\_modules/internal-slot": {

"version": "1.0.7",

"resolved": "https://registry.npmjs.org/internal-slot/-/internal-slot-1.0.7.tgz",

"integrity": "sha512-NGnrKwXzSms2qUUih/ILZ5JBqNTSa1+ZmP6flaIp6KmSElgE9qdndzS3cqjrDovwFdmwsGsLdeFgB6suw+1e9g==",

"dev": true,

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"hasown": "^2.0.0",

"side-channel": "^1.0.4"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/is-array-buffer": {

"version": "3.0.4",

"resolved": "https://registry.npmjs.org/is-array-buffer/-/is-array-buffer-3.0.4.tgz",

"integrity": "sha512-wcjaerHw0ydZwfhiKbXJWLDY8A7yV7KhjQOpb83hGgGfId/aQa4TOvwyzn2PuswW2gPCYEL/nEAiSVpdOj1lXw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"get-intrinsic": "^1.2.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-async-function": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/is-async-function/-/is-async-function-2.0.0.tgz",

"integrity": "sha512-Y1JXKrfykRJGdlDwdKlLpLyMIiWqWvuSd17TvZk68PLAOGOoF4Xyav1z0Xhoi+gCYjZVeC5SI+hYFOfvXmGRCA==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-bigint": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/is-bigint/-/is-bigint-1.0.4.tgz",

"integrity": "sha512-zB9CruMamjym81i2JZ3UMn54PKGsQzsJeo6xvN3HJJ4CAsQNB6iRutp2To77OfCNuoxspsIhzaPoO1zyCEhFOg==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-bigints": "^1.0.1"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-binary-path": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/is-binary-path/-/is-binary-path-2.1.0.tgz",

"integrity": "sha512-ZMERYes6pDydyuGidse7OsHxtbI7WVeUEozgR/g7rd0xUimYNlvZRE/K2MgZTjWy725IfelLeVcEM97mmtRGXw==",

"dev": true,

"license": "MIT",

"dependencies": {

"binary-extensions": "^2.0.0"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/is-boolean-object": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/is-boolean-object/-/is-boolean-object-1.1.2.tgz",

"integrity": "sha512-gDYaKHJmnj4aWxyj6YHyXVpdQawtVLHU5cb+eztPGczf6cjuTdwve5ZIEfgXqH4e57An1D1AKf8CZ3kYrQRqYA==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-callable": {

"version": "1.2.7",

"resolved": "https://registry.npmjs.org/is-callable/-/is-callable-1.2.7.tgz",

"integrity": "sha512-1BC0BVFhS/p0qtw6enp8e+8OD0UrK0oFLztSjNzhcKA3WDuJxxAPXzPuPtKkjEY9UUoEWlX/8fgKeu2S8i9JTA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-core-module": {

"version": "2.15.0",

"resolved": "https://registry.npmjs.org/is-core-module/-/is-core-module-2.15.0.tgz",

"integrity": "sha512-Dd+Lb2/zvk9SKy1TGCt1wFJFo/MWBPMX5x7KcvLajWTGuomczdQX61PvY5yK6SVACwpoexWo81IfFyoKY2QnTA==",

"dev": true,

"license": "MIT",

"dependencies": {

"hasown": "^2.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-data-view": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/is-data-view/-/is-data-view-1.0.1.tgz",

"integrity": "sha512-AHkaJrsUVW6wq6JS8y3JnM/GJF/9cf+k20+iDzlSaJrinEo5+7vRiteOSwBhHRiAyQATN1AmY4hwzxJKPmYf+w==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-typed-array": "^1.1.13"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-date-object": {

"version": "1.0.5",

"resolved": "https://registry.npmjs.org/is-date-object/-/is-date-object-1.0.5.tgz",

"integrity": "sha512-9YQaSxsAiSwcvS33MBk3wTCVnWK+HhF8VZR2jRxehM16QcVOdHqPn4VPHmRK4lSr38n9JriurInLcP90xsYNfQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-extglob": {

"version": "2.1.1",

"resolved": "https://registry.npmjs.org/is-extglob/-/is-extglob-2.1.1.tgz",

"integrity": "sha512-SbKbANkN603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfhGJ6QEDpOZIFkDtnq5JRxmvl3jsoQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/is-finalizationregistry": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/is-finalizationregistry/-/is-finalizationregistry-1.0.2.tgz",

"integrity": "sha512-0by5vtUJs8iFQb5TYUHHPudOR+qXYIMKtiUzvLIZITZUjknFmziyBJuLhVRc+Ds0dREFlskDNJKYIdIzu/9pfw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-fullwidth-code-point": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",

"integrity": "sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKclKdizYaJUuHA83RLjb7nSuGnddCHGv0hk+KY7BMAlsWeK4Ueg6EV6XQg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/is-generator-function": {

"version": "1.0.10",

"resolved": "https://registry.npmjs.org/is-generator-function/-/is-generator-function-1.0.10.tgz",

"integrity": "sha512-jsEjy9l3yiXEQ+PsXdmBwEPcOxaXWLspKdplFUVI9vq1iZgIekeC0L167qeu86czQaxed3q/Uzuw0swL0irL8A==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-glob": {

"version": "4.0.3",

"resolved": "https://registry.npmjs.org/is-glob/-/is-glob-4.0.3.tgz",

"integrity": "sha512-xelSayHH36ZgE7ZWhli7pW34hNbNl8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM+mXWOZhFkgZfxhLSnrwRr4elSSg==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-extglob": "^2.1.1"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/is-map": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/is-map/-/is-map-2.0.3.tgz",

"integrity": "sha512-1Qed0/Hr2m+YqxnM09CjA2d/i6YZNfF6R2oRAOj36eUdS6qIV/huPJNSEpKbupewFs+ZsJlxsjjPbc0/afW6Lw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-negative-zero": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/is-negative-zero/-/is-negative-zero-2.0.3.tgz",

"integrity": "sha512-5KoIu2Ngpyek75jXodFvnafB6DJgr3u8uuK0LEZJjrU19DrMD3EVERaR8sjz8CCGgpZvxPl9SuE1GMVPFHx1mw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-number": {

"version": "7.0.0",

"resolved": "https://registry.npmjs.org/is-number/-/is-number-7.0.0.tgz",

"integrity": "sha512-41Cifkg6e8TylSpdtTpeLVMqvSBEVzTttHvERD741+pnZ8ANv0004MRL43QKPDlK9cGvNp6NZWZUBlbGXYxxng==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.12.0"

}

},

"node\_modules/is-number-object": {

"version": "1.0.7",

"resolved": "https://registry.npmjs.org/is-number-object/-/is-number-object-1.0.7.tgz",

"integrity": "sha512-k1U0IRzLMo7ZlYIfzRu23Oh6MiIFasgpb9X76eqfFZAqwH44UI4KTBvBYIZ1dSL9ZzChTB9ShHfLkR4pdW5krQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-path-inside": {

"version": "3.0.3",

"resolved": "https://registry.npmjs.org/is-path-inside/-/is-path-inside-3.0.3.tgz",

"integrity": "sha512-Fd4gABb+ycGAmKou8eMftCupSir5lRxqf4aD/vd0cD2qc4HL07OjCeuHMr8Ro4CoMaeCKDB0/ECBOVWjTwUvPQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/is-regex": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/is-regex/-/is-regex-1.1.4.tgz",

"integrity": "sha512-kvRdxDsxZjhzUX07ZnLydzS1TU/TJlTUHHY4YLL87e37oUA49DfkLqgy+VjFocowy29cKvcSiu+kIv728jTTVg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-set": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/is-set/-/is-set-2.0.3.tgz",

"integrity": "sha512-iPAjerrse27/ygGLxw+EBR9agv9Y6uLeYVJMu+QNCoouJ1/1ri0mGrcWpfCqFZuzzx3WjtwxG098X+n4OuRkPg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-shared-array-buffer": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/is-shared-array-buffer/-/is-shared-array-buffer-1.0.3.tgz",

"integrity": "sha512-nA2hv5XIhLR3uVzDDfCIknerhx8XUKnstuOERPNNIinXG7v9u+ohXF67vxm4TPTEPU6lm61ZkwP3c9PCB97rhg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-string": {

"version": "1.0.7",

"resolved": "https://registry.npmjs.org/is-string/-/is-string-1.0.7.tgz",

"integrity": "sha512-tE2UXzivje6ofPW7l23cjDOMa09gb7xlAqG6jG5ej6uPV32TlWP3NKPigtaGeHNu9fohccRYvIiZMfOOnOYUtg==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-tostringtag": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-symbol": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/is-symbol/-/is-symbol-1.0.4.tgz",

"integrity": "sha512-C/CPBqKWnvdcxqIARxyOh4v1UUEOCHpgDa0WYgpKDFMszcrPcffg5uhwSgPCLD2WWxmq6isisz87tzT01tuGhg==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-symbols": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-typed-array": {

"version": "1.1.13",

"resolved": "https://registry.npmjs.org/is-typed-array/-/is-typed-array-1.1.13.tgz",

"integrity": "sha512-uZ25/bUAlUY5fR4OKT4rZQEBrzQWYV9ZJYGGsUmEJ6thodVJ1HX64ePQ6Z0qPWP+m+Uq6e9UugrE38jeYsDSMw==",

"dev": true,

"license": "MIT",

"dependencies": {

"which-typed-array": "^1.1.14"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-weakmap": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/is-weakmap/-/is-weakmap-2.0.2.tgz",

"integrity": "sha512-K5pXYOm9wqY1RgjpL3YTkF39tni1XajUIkawTLUo9EZEVUFga5gSQJF8nNS7ZwJQ02y+1YCNYcMh+HIf1ZqE+w==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-weakref": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/is-weakref/-/is-weakref-1.0.2.tgz",

"integrity": "sha512-qctsuLZmIQ0+vSSMfoVvyFe2+GSEvnmZ2ezTup1SBse9+twCCeial6EEi3Nc2KFcf6+qz2FBPnjXsk8xhKSaPQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/is-weakset": {

"version": "2.0.3",

"resolved": "https://registry.npmjs.org/is-weakset/-/is-weakset-2.0.3.tgz",

"integrity": "sha512-LvIm3/KWzS9oRFHugab7d+M/GcBXuXX5xZkzPmN+NxihdQlZUQ4dWuSV1xR/sq6upL1TJEDrfBgRepHFdBtSNQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"get-intrinsic": "^1.2.4"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/isarray": {

"version": "2.0.5",

"resolved": "https://registry.npmjs.org/isarray/-/isarray-2.0.5.tgz",

"integrity": "sha512-xHjhDr3cNBK0BzdUJSPXZntQUx/mwMS5Rw4A7lPJ90XGAO6ISP/ePDNuo0vhqOZU+UD5JoodwCAAoZQd3FeAKw==",

"dev": true,

"license": "MIT"

},

"node\_modules/isexe": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/isexe/-/isexe-2.0.0.tgz",

"integrity": "sha512-RHxMLp9lnKHGHRng9QFhRCMbYAcVpn69smSGcq3f36xjgVVWThj4qqLbTLlq7Ssj8B+fIQ1EuCEGI2lKsyQeIw==",

"dev": true,

"license": "ISC"

},

"node\_modules/iterator.prototype": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/iterator.prototype/-/iterator.prototype-1.1.2.tgz",

"integrity": "sha512-DR33HMMr8EzwuRL8Y9D3u2BMj8+RqSE850jfGu59kS7tbmPLzGkZmVSfyCFSDxuZiEY6Rzt3T2NA/qU+NwVj1w==",

"dev": true,

"license": "MIT",

"dependencies": {

"define-properties": "^1.2.1",

"get-intrinsic": "^1.2.1",

"has-symbols": "^1.0.3",

"reflect.getprototypeof": "^1.0.4",

"set-function-name": "^2.0.1"

}

},

"node\_modules/jackspeak": {

"version": "3.4.3",

"resolved": "https://registry.npmjs.org/jackspeak/-/jackspeak-3.4.3.tgz",

"integrity": "sha512-OGlZQpz2yfahA/Rd1Y8Cd9SIEsqvXkLVoSw/cgwhnhFMDbsQFeZYoJJ7bIZBS9BcamUW96asq/npPWugM+RQBw==",

"dev": true,

"license": "BlueOak-1.0.0",

"dependencies": {

"@isaacs/cliui": "^8.0.2"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

},

"optionalDependencies": {

"@pkgjs/parseargs": "^0.11.0"

}

},

"node\_modules/jiti": {

"version": "1.21.6",

"resolved": "https://registry.npmjs.org/jiti/-/jiti-1.21.6.tgz",

"integrity": "sha512-2yTgeWTWzMWkHu6Jp9NKgePDaYHbntiwvYuuJLbbN9vl7DC9DvXKOB2BC3ZZ92D3cvV/aflH0osDfwpHepQ53w==",

"dev": true,

"license": "MIT",

"bin": {

"jiti": "bin/jiti.js"

}

},

"node\_modules/js-tokens": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/js-tokens/-/js-tokens-4.0.0.tgz",

"integrity": "sha512-RdJUflcE3cUzKiMqQgsCu06FPu9UdIJO0beYbPhHN4k6apgJtifcoCtT9bcxOpYBtpD2kCM6Sbzg4CausW/PKQ==",

"license": "MIT"

},

"node\_modules/js-yaml": {

"version": "4.1.0",

"resolved": "https://registry.npmjs.org/js-yaml/-/js-yaml-4.1.0.tgz",

"integrity": "sha512-wpxZs9NoxZaJESJGIZTyDEaYpl0FKSA+FB9aJiyemKhMwkxQg63h4T1KJgUGHpTqPDNRcmmYLugrRjJlBtWvRA==",

"dev": true,

"license": "MIT",

"dependencies": {

"argparse": "^2.0.1"

},

"bin": {

"js-yaml": "bin/js-yaml.js"

}

},

"node\_modules/jsesc": {

"version": "2.5.2",

"resolved": "https://registry.npmjs.org/jsesc/-/jsesc-2.5.2.tgz",

"integrity": "sha512-OYu7XEzjkCQ3C5Ps3QIZsQfNpqoJyZZA99wd9aWd05NCtC5pWOkShK2mkL6HXQR6/Cy2lbNdPlZBpuQHXE63gA==",

"dev": true,

"license": "MIT",

"bin": {

"jsesc": "bin/jsesc"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/json-buffer": {

"version": "3.0.1",

"resolved": "https://registry.npmjs.org/json-buffer/-/json-buffer-3.0.1.tgz",

"integrity": "sha512-4bV5BfR2mqfQTJm+V5tPPdf+ZpuhiIvTuAB5g8kcrXOZpTT/QwwVRWBywX1ozr6lEuPdbHxwaJlm9G6mI2sfSQ==",

"dev": true,

"license": "MIT"

},

"node\_modules/json-schema-traverse": {

"version": "0.4.1",

"resolved": "https://registry.npmjs.org/json-schema-traverse/-/json-schema-traverse-0.4.1.tgz",

"integrity": "sha512-xbbCH5dCYU5T8LcEhhuh7HJ88HXuW3qsI3Y0zOZFKfZEHcpWiHU/Jxzk629Brsab/mMiHQti9wMP+845RPe3Vg==",

"dev": true,

"license": "MIT"

},

"node\_modules/json-stable-stringify-without-jsonify": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/json-stable-stringify-without-jsonify/-/json-stable-stringify-without-jsonify-1.0.1.tgz",

"integrity": "sha512-Bdboy+l7tA3OGW6FjyFHWkP5LuByj1Tk33Ljyq0axyzdk9//JSi2u3fP1QSmd1KNwq6VOKYGlAu87CisVir6Pw==",

"dev": true,

"license": "MIT"

},

"node\_modules/json5": {

"version": "2.2.3",

"resolved": "https://registry.npmjs.org/json5/-/json5-2.2.3.tgz",

"integrity": "sha512-XmOWe7eyHYH14cLdVPoyg+GOH3rYX++KpzrylJwSW98t3Nk+U8XOl8FWKOgwtzdb8lXGf6zYwDUzeHMWfxasyg==",

"dev": true,

"license": "MIT",

"bin": {

"json5": "lib/cli.js"

},

"engines": {

"node": ">=6"

}

},

"node\_modules/jsx-ast-utils": {

"version": "3.3.5",

"resolved": "https://registry.npmjs.org/jsx-ast-utils/-/jsx-ast-utils-3.3.5.tgz",

"integrity": "sha512-ZZow9HBI5O6EPgSJLUb8n2NKgmVWTwCvHGwFuJlMjvLFqlGG6pjirPhtdsseaLZjSibD8eegzmYpUZwoIlj2cQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"array-includes": "^3.1.6",

"array.prototype.flat": "^1.3.1",

"object.assign": "^4.1.4",

"object.values": "^1.1.6"

},

"engines": {

"node": ">=4.0"

}

},

"node\_modules/keyv": {

"version": "4.5.4",

"resolved": "https://registry.npmjs.org/keyv/-/keyv-4.5.4.tgz",

"integrity": "sha512-oxVHkHR/EJf2CNXnWxRLW6mg7JyCCUcG0DtEGmL2ctUo1PNTin1PUil+r/+4r5MpVgC/fn1kjsx7mjSujKqIpw==",

"dev": true,

"license": "MIT",

"dependencies": {

"json-buffer": "3.0.1"

}

},

"node\_modules/levn": {

"version": "0.4.1",

"resolved": "https://registry.npmjs.org/levn/-/levn-0.4.1.tgz",

"integrity": "sha512-+bT2uH4E5LGE7h/n3evcS/sQlJXCpIp6ym8OWJ5eV6+67Dsql/LaaT7qJBAt2rzfoa/5QBGBhxDix1dMt2kQKQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"prelude-ls": "^1.2.1",

"type-check": "~0.4.0"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/lilconfig": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/lilconfig/-/lilconfig-2.1.0.tgz",

"integrity": "sha512-utWOt/GHzuUxnLKxB6dk81RoOeoNeHgbrXiuGk4yyF5qlRz+iIVWu56E2fqGHFrXz0QNUhLB/8nKqvRH66JKGQ==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=10"

}

},

"node\_modules/lines-and-columns": {

"version": "1.2.4",

"resolved": "https://registry.npmjs.org/lines-and-columns/-/lines-and-columns-1.2.4.tgz",

"integrity": "sha512-7ylylesZQ/PV29jhEDl3Ufjo6ZX7gCqJr5F7PKrqc93v7fzSymt1BpwEU8nAUXs8qzzvqhbjhK5QZg6Mt/HkBg==",

"dev": true,

"license": "MIT"

},

"node\_modules/locate-path": {

"version": "6.0.0",

"resolved": "https://registry.npmjs.org/locate-path/-/locate-path-6.0.0.tgz",

"integrity": "sha512-iPZK6eYjbxRu3uB4/WZ3EsEIMJFMqAoopl3R+zuq0UjcAm/MO6KCweDgPfP3elTztoKP3KtnVHxTn2NHBSDVUw==",

"dev": true,

"license": "MIT",

"dependencies": {

"p-locate": "^5.0.0"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/lodash.merge": {

"version": "4.6.2",

"resolved": "https://registry.npmjs.org/lodash.merge/-/lodash.merge-4.6.2.tgz",

"integrity": "sha512-0KpjqXRVvrYyCsX1swR/XTK0va6VQkQM6MNo7PqW77ByjAhoARA8EfrP1N4+KlKj8YS0ZUCtRT/YUuhyYDujIQ==",

"dev": true,

"license": "MIT"

},

"node\_modules/loose-envify": {

"version": "1.4.0",

"resolved": "https://registry.npmjs.org/loose-envify/-/loose-envify-1.4.0.tgz",

"integrity": "sha512-lyuxPGr/Wfhrlem2CL/UcnUc1zcqKAImBDzukY7Y5F/yQiNdko6+fRLevlw1HgMySw7f611UIY408EtxRSoK3Q==",

"license": "MIT",

"dependencies": {

"js-tokens": "^3.0.0 || ^4.0.0"

},

"bin": {

"loose-envify": "cli.js"

}

},

"node\_modules/lru-cache": {

"version": "5.1.1",

"resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-5.1.1.tgz",

"integrity": "sha512-KpNARQA3Iwv+jTA0utUVVbrh+Jlrr1Fv0e56GGzAFOXN7dk/FviaDW8LHmK52DlcH4WP2n6gI8vN1aesBFgo9w==",

"dev": true,

"license": "ISC",

"dependencies": {

"yallist": "^3.0.2"

}

},

"node\_modules/merge2": {

"version": "1.4.1",

"resolved": "https://registry.npmjs.org/merge2/-/merge2-1.4.1.tgz",

"integrity": "sha512-8q7VEgMJW4J8tcfVPy8g09NcQwZdbwFEqhe/WZkoIzjn/3TGDwtOCYtXGxA3O8tPzpczCCDgv+P2P5y00ZJOOg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 8"

}

},

"node\_modules/micromatch": {

"version": "4.0.7",

"resolved": "https://registry.npmjs.org/micromatch/-/micromatch-4.0.7.tgz",

"integrity": "sha512-LPP/3KorzCwBxfeUuZmaR6bG2kdeHSbe0P2tY3FLRU4vYrjYz5hI4QZwV0njUx3jeuKe67YukQ1LSPZBKDqO/Q==",

"dev": true,

"license": "MIT",

"dependencies": {

"braces": "^3.0.3",

"picomatch": "^2.3.1"

},

"engines": {

"node": ">=8.6"

}

},

"node\_modules/mime-db": {

"version": "1.52.0",

"resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",

"integrity": "sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/mime-types": {

"version": "2.1.35",

"resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",

"integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",

"license": "MIT",

"dependencies": {

"mime-db": "1.52.0"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/minimatch": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",

"integrity": "sha512-J7p63hRiAjw1NDEww1W7i37+ByIrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==",

"dev": true,

"license": "ISC",

"dependencies": {

"brace-expansion": "^1.1.7"

},

"engines": {

"node": "\*"

}

},

"node\_modules/minipass": {

"version": "7.1.2",

"resolved": "https://registry.npmjs.org/minipass/-/minipass-7.1.2.tgz",

"integrity": "sha512-qOOzS1cBTWYF4BH8fVePDBOO9iptMnGUEZwNc/cMWnTV2nVLZ7VoNWEPHkYczZA0pdoA7dl6e7FL659nX9S2aw==",

"dev": true,

"license": "ISC",

"engines": {

"node": ">=16 || 14 >=14.17"

}

},

"node\_modules/ms": {

"version": "2.1.2",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.2.tgz",

"integrity": "sha512-sGkPx+VjMtmA6MX27oA4FBFELFCZZ4S4XqeGOXCv68tT+jb3vk/RyaKWP0PTKyWtmLSM0b+adUTEvbs1PEaH2w==",

"dev": true,

"license": "MIT"

},

"node\_modules/mz": {

"version": "2.7.0",

"resolved": "https://registry.npmjs.org/mz/-/mz-2.7.0.tgz",

"integrity": "sha512-z81GNO7nnYMEhrGh9LeymoE4+Yr0Wn5McHIZMK5cfQCl+NDX08sCZgUc9/6MHni9IWuFLm1Z3HTCXu2z9fN62Q==",

"dev": true,

"license": "MIT",

"dependencies": {

"any-promise": "^1.0.0",

"object-assign": "^4.0.1",

"thenify-all": "^1.0.0"

}

},

"node\_modules/nanoid": {

"version": "3.3.7",

"resolved": "https://registry.npmjs.org/nanoid/-/nanoid-3.3.7.tgz",

"integrity": "sha512-eSRppjcPIatRIMC1U6UngP8XFcz8MQWGQdt1MTBQ7NaAmvXDfvNxbvWV3x2y6CdEUciCSsDHDQZbhYaB8QEo2g==",

"dev": true,

"funding": [

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"bin": {

"nanoid": "bin/nanoid.cjs"

},

"engines": {

"node": "^10 || ^12 || ^13.7 || ^14 || >=15.0.1"

}

},

"node\_modules/natural-compare": {

"version": "1.4.0",

"resolved": "https://registry.npmjs.org/natural-compare/-/natural-compare-1.4.0.tgz",

"integrity": "sha512-OWND8ei3VtNC9h7V60qff3SVobHr996CTwgxubgyQYEpg290h9J0buyECNNJexkFm5sOajh5G116RYA1c8ZMSw==",

"dev": true,

"license": "MIT"

},

"node\_modules/node-releases": {

"version": "2.0.18",

"resolved": "https://registry.npmjs.org/node-releases/-/node-releases-2.0.18.tgz",

"integrity": "sha512-d9VeXT4SJ7ZeOqGX6R5EM022wpL+eWPooLI+5UpWn2jCT1aosUQEhQP214x33Wkwx3JQMvIm+tIoVOdodFS40g==",

"dev": true,

"license": "MIT"

},

"node\_modules/normalize-path": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/normalize-path/-/normalize-path-3.0.0.tgz",

"integrity": "sha512-6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6IKbQkY2Pnb/mD4WYojCRwcwLA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/normalize-range": {

"version": "0.1.2",

"resolved": "https://registry.npmjs.org/normalize-range/-/normalize-range-0.1.2.tgz",

"integrity": "sha512-bdok/XvKII3nUpklnV6P2hxtMNrCboOjAcyBuQnWEhO665FwrSNRxU+AqpsyvO6LgGYPspN+lu5CLtw4jPRKNA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/object-assign": {

"version": "4.1.1",

"resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",

"integrity": "sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/object-hash": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/object-hash/-/object-hash-3.0.0.tgz",

"integrity": "sha512-RSn9F68PjH9HqtltsSnqYC1XXoWe9Bju5+213R98cNGttag9q9yAOTzdbsqvIa7aNm5WffBZFpWYr2aWrklWAw==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 6"

}

},

"node\_modules/object-inspect": {

"version": "1.13.2",

"resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.2.tgz",

"integrity": "sha512-IRZSRuzJiynemAXPYtPe5BoI/RESNYR7TYm50MC5Mqbd3Jmw5y790sErYw3V6SryFJD64b74qQQs9wn5Bg/k3g==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/object-keys": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/object-keys/-/object-keys-1.1.1.tgz",

"integrity": "sha512-NuAESUOUMrlIXOfHKzD6bpPu3tYt3xvjNdRIQ+FeT0lNb4K8WR70CaDxhuNguS2XG+GjkyMwOzsN5ZktImfhLA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/object.assign": {

"version": "4.1.5",

"resolved": "https://registry.npmjs.org/object.assign/-/object.assign-4.1.5.tgz",

"integrity": "sha512-byy+U7gp+FVwmyzKPYhW2h5l3crpmGsxl7X2s8y43IgxvG4g3QZ6CffDtsNQy1WsmZpQbO+ybo0AlW7TY6DcBQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.5",

"define-properties": "^1.2.1",

"has-symbols": "^1.0.3",

"object-keys": "^1.1.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/object.entries": {

"version": "1.1.8",

"resolved": "https://registry.npmjs.org/object.entries/-/object.entries-1.1.8.tgz",

"integrity": "sha512-cmopxi8VwRIAw/fkijJohSfpef5PdN0pMQJN6VC/ZKvn0LIknWD8KtgY6KlQdEc4tIjcQ3HxSMmnvtzIscdaYQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/object.fromentries": {

"version": "2.0.8",

"resolved": "https://registry.npmjs.org/object.fromentries/-/object.fromentries-2.0.8.tgz",

"integrity": "sha512-k6E21FzySsSK5a21KRADBd/NGneRegFO5pLHfdQLpRDETUNJueLXs3WCzyQ3tFRDYgbq3KHGXfTbi2bs8WQ6rQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.2",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/object.values": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/object.values/-/object.values-1.2.0.tgz",

"integrity": "sha512-yBYjY9QX2hnRmZHAjG/f13MzmBzxzYgQhFrke06TTyKY5zSTEqkOeukBzIdVA3j3ulu8Qa3MbVFShV7T2RmGtQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/once": {

"version": "1.4.0",

"resolved": "https://registry.npmjs.org/once/-/once-1.4.0.tgz",

"integrity": "sha512-lNaJgI+2Q5URQBkccEKHTQOPaXdUxnZZElQTZY0MFUAuaEqe1E+Nyvgdz/aIyNi6Z9MzO5dv1H8n58/GELp3+w==",

"dev": true,

"license": "ISC",

"dependencies": {

"wrappy": "1"

}

},

"node\_modules/optionator": {

"version": "0.9.4",

"resolved": "https://registry.npmjs.org/optionator/-/optionator-0.9.4.tgz",

"integrity": "sha512-6IpQ7mKUxRcZNLIObR0hz7lxsapSSIYNZJwXPGeF0mTVqGKFIXj1DQcMoT22S3ROcLyY/rz0PWaWZ9ayWmad9g==",

"dev": true,

"license": "MIT",

"dependencies": {

"deep-is": "^0.1.3",

"fast-levenshtein": "^2.0.6",

"levn": "^0.4.1",

"prelude-ls": "^1.2.1",

"type-check": "^0.4.0",

"word-wrap": "^1.2.5"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/p-limit": {

"version": "3.1.0",

"resolved": "https://registry.npmjs.org/p-limit/-/p-limit-3.1.0.tgz",

"integrity": "sha512-TYOanM3wGwNGsZN2cVTYPArw454xnXj5qmWF1bEoAc4+cU/ol7GVh7odevjp1FNHduHc3KZMcFduxU5Xc6uJRQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"yocto-queue": "^0.1.0"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/p-locate": {

"version": "5.0.0",

"resolved": "https://registry.npmjs.org/p-locate/-/p-locate-5.0.0.tgz",

"integrity": "sha512-LaNjtRWUBY++zB5nE/NwcaoMylSPk+S+ZHNB1TzdbMJMny6dynpAGt7X/tl/QYq3TIeE6nxHppbo2LGymrG5Pw==",

"dev": true,

"license": "MIT",

"dependencies": {

"p-limit": "^3.0.2"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/package-json-from-dist": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/package-json-from-dist/-/package-json-from-dist-1.0.0.tgz",

"integrity": "sha512-dATvCeZN/8wQsGywez1mzHtTlP22H8OEfPrVMLNr4/eGa+ijtLn/6M5f0dY8UKNrC2O9UCU6SSoG3qRKnt7STw==",

"dev": true,

"license": "BlueOak-1.0.0"

},

"node\_modules/parent-module": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/parent-module/-/parent-module-1.0.1.tgz",

"integrity": "sha512-GQ2EWRpQV8/o+Aw8YqtfZZPfNRWZYkbidE9k5rpl/hC3vtHHBfGm2Ifi6qWV+coDGkrUKZAxE3Lot5kcsRlh+g==",

"dev": true,

"license": "MIT",

"dependencies": {

"callsites": "^3.0.0"

},

"engines": {

"node": ">=6"

}

},

"node\_modules/path-exists": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/path-exists/-/path-exists-4.0.0.tgz",

"integrity": "sha512-ak9Qy5Q7jYb2Wwcey5Fpvg2KoAc/ZIhLSLOSBmRmygPsGwkVVt0fZa0qrtMz+m6tJTAHfZQ8FnmB4MG4LWy7/w==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/path-is-absolute": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/path-is-absolute/-/path-is-absolute-1.0.1.tgz",

"integrity": "sha512-AVbw3UJ2e9bq64vSaS9Am0fje1Pa8pbGqTTsmXfaIiMpnr5DlDhfJOuLj9Sf95ZPVDAUerDfEk88MPmPe7UCQg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/path-key": {

"version": "3.1.1",

"resolved": "https://registry.npmjs.org/path-key/-/path-key-3.1.1.tgz",

"integrity": "sha512-ojmeN0qd+y0jszEtoY48r0Peq5dwMEkIlCOu6Q5f41lfkswXuKtYrhgoTpLnyIcHm24Uhqx+5Tqm2InSwLhE6Q==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/path-parse": {

"version": "1.0.7",

"resolved": "https://registry.npmjs.org/path-parse/-/path-parse-1.0.7.tgz",

"integrity": "sha512-LDJzPVEEEPR+y48z93A0Ed0yXb8pAByGWo/k5YYdYgpY2/2EsOsksJrq7lOHxryrVOn1ejG6oAp8ahvOIQD8sw==",

"dev": true,

"license": "MIT"

},

"node\_modules/path-scurry": {

"version": "1.11.1",

"resolved": "https://registry.npmjs.org/path-scurry/-/path-scurry-1.11.1.tgz",

"integrity": "sha512-Xa4Nw17FS9ApQFJ9umLiJS4orGjm7ZzwUrwamcGQuHSzDyth9boKDaycYdDcZDuqYATXw4HFXgaqWTctW/v1HA==",

"dev": true,

"license": "BlueOak-1.0.0",

"dependencies": {

"lru-cache": "^10.2.0",

"minipass": "^5.0.0 || ^6.0.2 || ^7.0.0"

},

"engines": {

"node": ">=16 || 14 >=14.18"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/path-scurry/node\_modules/lru-cache": {

"version": "10.4.3",

"resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-10.4.3.tgz",

"integrity": "sha512-JNAzZcXrCt42VGLuYz0zfAzDfAvJWW6AfYlDBQyDV5DClI2m5sAmK+OIO7s59XfsRsWHp02jAJrRadPRGTt6SQ==",

"dev": true,

"license": "ISC"

},

"node\_modules/picocolors": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/picocolors/-/picocolors-1.0.1.tgz",

"integrity": "sha512-anP1Z8qwhkbmu7MFP5iTt+wQKXgwzf7zTyGlcdzabySa9vd0Xt392U0rVmz9poOaBj0uHJKyyo9/upk0HrEQew==",

"dev": true,

"license": "ISC"

},

"node\_modules/picomatch": {

"version": "2.3.1",

"resolved": "https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz",

"integrity": "sha512-JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fsIvwaStHhAWJEeVd1o1QD80CU6+ZdEXXSLbSsuLwJjkCBWqRQUVA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8.6"

},

"funding": {

"url": "https://github.com/sponsors/jonschlinkert"

}

},

"node\_modules/pify": {

"version": "2.3.0",

"resolved": "https://registry.npmjs.org/pify/-/pify-2.3.0.tgz",

"integrity": "sha512-udgsAY+fTnvv7kI7aaxbqwWNb0AHiB0qBO89PZKPkoTmGOgdbrHDKD+0B2X4uTfJ/FT1R09r9gTsjUjNJotuog==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/pirates": {

"version": "4.0.6",

"resolved": "https://registry.npmjs.org/pirates/-/pirates-4.0.6.tgz",

"integrity": "sha512-saLsH7WeYYPiD25LDuLRRY/i+6HaPYr6G1OUlN39otzkSTxKnubR9RTxS3/Kk50s1g2JTgFwWQDQyplC5/SHZg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 6"

}

},

"node\_modules/possible-typed-array-names": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/possible-typed-array-names/-/possible-typed-array-names-1.0.0.tgz",

"integrity": "sha512-d7Uw+eZoloe0EHDIYoe+bQ5WXnGMOpmiZFTuMWCwpjzzkL2nTjcKiAk4hh8TjnGye2TwWOk3UXucZ+3rbmBa8Q==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/postcss": {

"version": "8.4.41",

"resolved": "https://registry.npmjs.org/postcss/-/postcss-8.4.41.tgz",

"integrity": "sha512-TesUflQ0WKZqAvg52PWL6kHgLKP6xB6heTOdoYM0Wt2UHyxNa4K25EZZMgKns3BH1RLVbZCREPpLY0rhnNoHVQ==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/postcss/"

},

{

"type": "tidelift",

"url": "https://tidelift.com/funding/github/npm/postcss"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"dependencies": {

"nanoid": "^3.3.7",

"picocolors": "^1.0.1",

"source-map-js": "^1.2.0"

},

"engines": {

"node": "^10 || ^12 || >=14"

}

},

"node\_modules/postcss-import": {

"version": "15.1.0",

"resolved": "https://registry.npmjs.org/postcss-import/-/postcss-import-15.1.0.tgz",

"integrity": "sha512-hpr+J05B2FVYUAXHeK1YyI267J/dDDhMU6B6civm8hSY1jYJnBXxzKDKDswzJmtLHryrjhnDjqqp/49t8FALew==",

"dev": true,

"license": "MIT",

"dependencies": {

"postcss-value-parser": "^4.0.0",

"read-cache": "^1.0.0",

"resolve": "^1.1.7"

},

"engines": {

"node": ">=14.0.0"

},

"peerDependencies": {

"postcss": "^8.0.0"

}

},

"node\_modules/postcss-import/node\_modules/resolve": {

"version": "1.22.8",

"resolved": "https://registry.npmjs.org/resolve/-/resolve-1.22.8.tgz",

"integrity": "sha512-oKWePCxqpd6FlLvGV1VU0x7bkPmmCNolxzjMf4NczoDnQcIWrAF+cPtZn5i6n+RfD2d9i0tzpKnG6Yk168yIyw==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-core-module": "^2.13.0",

"path-parse": "^1.0.7",

"supports-preserve-symlinks-flag": "^1.0.0"

},

"bin": {

"resolve": "bin/resolve"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/postcss-js": {

"version": "4.0.1",

"resolved": "https://registry.npmjs.org/postcss-js/-/postcss-js-4.0.1.tgz",

"integrity": "sha512-dDLF8pEO191hJMtlHFPRa8xsizHaM82MLfNkUHdUtVEV3tgTp5oj+8qbEqYM57SLfc74KSbw//4SeJma2LRVIw==",

"dev": true,

"license": "MIT",

"dependencies": {

"camelcase-css": "^2.0.1"

},

"engines": {

"node": "^12 || ^14 || >= 16"

},

"funding": {

"type": "opencollective",

"url": "https://opencollective.com/postcss/"

},

"peerDependencies": {

"postcss": "^8.4.21"

}

},

"node\_modules/postcss-load-config": {

"version": "4.0.2",

"resolved": "https://registry.npmjs.org/postcss-load-config/-/postcss-load-config-4.0.2.tgz",

"integrity": "sha512-bSVhyJGL00wMVoPUzAVAnbEoWyqRxkjv64tUl427SKnPrENtq6hJwUojroMz2VB+Q1edmi4IfrAPpami5VVgMQ==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/postcss/"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"dependencies": {

"lilconfig": "^3.0.0",

"yaml": "^2.3.4"

},

"engines": {

"node": ">= 14"

},

"peerDependencies": {

"postcss": ">=8.0.9",

"ts-node": ">=9.0.0"

},

"peerDependenciesMeta": {

"postcss": {

"optional": true

},

"ts-node": {

"optional": true

}

}

},

"node\_modules/postcss-load-config/node\_modules/lilconfig": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/lilconfig/-/lilconfig-3.1.2.tgz",

"integrity": "sha512-eop+wDAvpItUys0FWkHIKeC9ybYrTGbU41U5K7+bttZZeohvnY7M9dZ5kB21GNWiFT2q1OoPTvncPCgSOVO5ow==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=14"

},

"funding": {

"url": "https://github.com/sponsors/antonk52"

}

},

"node\_modules/postcss-nested": {

"version": "6.2.0",

"resolved": "https://registry.npmjs.org/postcss-nested/-/postcss-nested-6.2.0.tgz",

"integrity": "sha512-HQbt28KulC5AJzG+cZtj9kvKB93CFCdLvog1WFLf1D+xmMvPGlBstkpTEZfK5+AN9hfJocyBFCNiqyS48bpgzQ==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/postcss/"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"dependencies": {

"postcss-selector-parser": "^6.1.1"

},

"engines": {

"node": ">=12.0"

},

"peerDependencies": {

"postcss": "^8.2.14"

}

},

"node\_modules/postcss-selector-parser": {

"version": "6.1.1",

"resolved": "https://registry.npmjs.org/postcss-selector-parser/-/postcss-selector-parser-6.1.1.tgz",

"integrity": "sha512-b4dlw/9V8A71rLIDsSwVmak9z2DuBUB7CA1/wSdelNEzqsjoSPeADTWNO09lpH49Diy3/JIZ2bSPB1dI3LJCHg==",

"dev": true,

"license": "MIT",

"dependencies": {

"cssesc": "^3.0.0",

"util-deprecate": "^1.0.2"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/postcss-value-parser": {

"version": "4.2.0",

"resolved": "https://registry.npmjs.org/postcss-value-parser/-/postcss-value-parser-4.2.0.tgz",

"integrity": "sha512-1NNCs6uurfkVbeXG4S8JFT9t19m45ICnif8zWLd5oPSZ50QnwMfK+H3jv408d4jw/7Bttv5axS5IiHoLaVNHeQ==",

"dev": true,

"license": "MIT"

},

"node\_modules/prelude-ls": {

"version": "1.2.1",

"resolved": "https://registry.npmjs.org/prelude-ls/-/prelude-ls-1.2.1.tgz",

"integrity": "sha512-vkcDPrRZo1QZLbn5RLGPpg/WmIQ65qoWWhcGKf/b5eplkkarX0m9z8ppCat4mlOqUsWpyNuYgO3VRyrYHSzX5g==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/prop-types": {

"version": "15.8.1",

"resolved": "https://registry.npmjs.org/prop-types/-/prop-types-15.8.1.tgz",

"integrity": "sha512-oj87CgZICdulUohogVAR7AjlC0327U4el4L6eAvOqCeudMDVU0NThNaV+b9Df4dXgSP1gXMTnPdhfe/2qDH5cg==",

"dev": true,

"license": "MIT",

"dependencies": {

"loose-envify": "^1.4.0",

"object-assign": "^4.1.1",

"react-is": "^16.13.1"

}

},

"node\_modules/proxy-from-env": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/proxy-from-env/-/proxy-from-env-1.1.0.tgz",

"integrity": "sha512-D+zkORCbA9f1tdWRK0RaCR3GPv50cMxcrz4X8k5LTSUD1Dkw47mKJEZQNunItRTkWwgtaUSo1RVFRIG9ZXiFYg==",

"license": "MIT"

},

"node\_modules/punycode": {

"version": "2.3.1",

"resolved": "https://registry.npmjs.org/punycode/-/punycode-2.3.1.tgz",

"integrity": "sha512-vYt7UD1U9Wg6138shLtLOvdAu+8DsC/ilFtEVHcH+wydcSpNE20AfSOduf6MkRFahL5FY7X1oU7nKVZFtfq8Fg==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=6"

}

},

"node\_modules/queue-microtask": {

"version": "1.2.3",

"resolved": "https://registry.npmjs.org/queue-microtask/-/queue-microtask-1.2.3.tgz",

"integrity": "sha512-NuaNSa6flKT5JaSYQzJok04JzTL1CA6aGhv5rfLW3PgqA+M2ChpZQnAC8h8i4ZFkBS8X5RqkDBHA7r4hej3K9A==",

"dev": true,

"funding": [

{

"type": "github",

"url": "https://github.com/sponsors/feross"

},

{

"type": "patreon",

"url": "https://www.patreon.com/feross"

},

{

"type": "consulting",

"url": "https://feross.org/support"

}

],

"license": "MIT"

},

"node\_modules/react": {

"version": "18.3.1",

"resolved": "https://registry.npmjs.org/react/-/react-18.3.1.tgz",

"integrity": "sha512-wS+hAgJShR0KhEvPJArfuPVN1+Hz1t0Y6n5jLrGQbkb4urgPE/0Rve+1kMB1v/oWgHgm4WIcV+i7F2pTVj+2iQ==",

"license": "MIT",

"dependencies": {

"loose-envify": "^1.1.0"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/react-dom": {

"version": "18.3.1",

"resolved": "https://registry.npmjs.org/react-dom/-/react-dom-18.3.1.tgz",

"integrity": "sha512-5m4nQKp+rZRb09LNH59GM4BxTh9251/ylbKIbpe7TpGxfJ+9kv6BLkLBXIjjspbgbnIBNqlI23tRnTWT0snUIw==",

"license": "MIT",

"dependencies": {

"loose-envify": "^1.1.0",

"scheduler": "^0.23.2"

},

"peerDependencies": {

"react": "^18.3.1"

}

},

"node\_modules/react-is": {

"version": "16.13.1",

"resolved": "https://registry.npmjs.org/react-is/-/react-is-16.13.1.tgz",

"integrity": "sha512-24e6ynE2H+OKt4kqsOvNd8kBpV65zoxbA4BVsEOB3ARVWQki/DHzaUoC5KuON/BiccDaCCTZBuOcfZs70kR8bQ==",

"dev": true,

"license": "MIT"

},

"node\_modules/react-refresh": {

"version": "0.14.2",

"resolved": "https://registry.npmjs.org/react-refresh/-/react-refresh-0.14.2.tgz",

"integrity": "sha512-jCvmsr+1IUSMUyzOkRcvnVbX3ZYC6g9TDrDbFuFmRDq7PD4yaGbLKNQL6k2jnArV8hjYxh7hVhAZB6s9HDGpZA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/react-router": {

"version": "6.26.0",

"resolved": "https://registry.npmjs.org/react-router/-/react-router-6.26.0.tgz",

"integrity": "sha512-wVQq0/iFYd3iZ9H2l3N3k4PL8EEHcb0XlU2Na8nEwmiXgIUElEH6gaJDtUQxJ+JFzmIXaQjfdpcGWaM6IoQGxg==",

"license": "MIT",

"dependencies": {

"@remix-run/router": "1.19.0"

},

"engines": {

"node": ">=14.0.0"

},

"peerDependencies": {

"react": ">=16.8"

}

},

"node\_modules/react-router-dom": {

"version": "6.26.0",

"resolved": "https://registry.npmjs.org/react-router-dom/-/react-router-dom-6.26.0.tgz",

"integrity": "sha512-RRGUIiDtLrkX3uYcFiCIxKFWMcWQGMojpYZfcstc63A1+sSnVgILGIm9gNUA6na3Fm1QuPGSBQH2EMbAZOnMsQ==",

"license": "MIT",

"dependencies": {

"@remix-run/router": "1.19.0",

"react-router": "6.26.0"

},

"engines": {

"node": ">=14.0.0"

},

"peerDependencies": {

"react": ">=16.8",

"react-dom": ">=16.8"

}

},

"node\_modules/react-toastify": {

"version": "10.0.5",

"resolved": "https://registry.npmjs.org/react-toastify/-/react-toastify-10.0.5.tgz",

"integrity": "sha512-mNKt2jBXJg4O7pSdbNUfDdTsK9FIdikfsIE/yUCxbAEXl4HMyJaivrVFcn3Elvt5xvCQYhUZm+hqTIu1UXM3Pw==",

"license": "MIT",

"dependencies": {

"clsx": "^2.1.0"

},

"peerDependencies": {

"react": ">=18",

"react-dom": ">=18"

}

},

"node\_modules/read-cache": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/read-cache/-/read-cache-1.0.0.tgz",

"integrity": "sha512-Owdv/Ft7IjOgm/i0xvNDZ1LrRANRfew4b2prF3OWMQLxLfu3bS8FVhCsrSCMK4lR56Y9ya+AThoTpDCTxCmpRA==",

"dev": true,

"license": "MIT",

"dependencies": {

"pify": "^2.3.0"

}

},

"node\_modules/readdirp": {

"version": "3.6.0",

"resolved": "https://registry.npmjs.org/readdirp/-/readdirp-3.6.0.tgz",

"integrity": "sha512-hOS089on8RduqdbhvQ5Z37A0ESjsqz6qnRcffsMU3495FuTdqSm+7bhJ29JvIOsBDEEnan5DPu9t3To9VRlMzA==",

"dev": true,

"license": "MIT",

"dependencies": {

"picomatch": "^2.2.1"

},

"engines": {

"node": ">=8.10.0"

}

},

"node\_modules/reflect.getprototypeof": {

"version": "1.0.6",

"resolved": "https://registry.npmjs.org/reflect.getprototypeof/-/reflect.getprototypeof-1.0.6.tgz",

"integrity": "sha512-fmfw4XgoDke3kdI6h4xcUz1dG8uaiv5q9gcEwLS4Pnth2kxT+GZ7YehS1JTMGBQmtV7Y4GFGbs2re2NqhdozUg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.1",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.4",

"globalthis": "^1.0.3",

"which-builtin-type": "^1.1.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/regexp.prototype.flags": {

"version": "1.5.2",

"resolved": "https://registry.npmjs.org/regexp.prototype.flags/-/regexp.prototype.flags-1.5.2.tgz",

"integrity": "sha512-NcDiDkTLuPR+++OCKB0nWafEmhg/Da8aUPLPMQbK+bxKKCm1/S5he+AqYa4PlMCVBalb4/yxIRub6qkEx5yJbw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.6",

"define-properties": "^1.2.1",

"es-errors": "^1.3.0",

"set-function-name": "^2.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/resolve": {

"version": "2.0.0-next.5",

"resolved": "https://registry.npmjs.org/resolve/-/resolve-2.0.0-next.5.tgz",

"integrity": "sha512-U7WjGVG9sH8tvjW5SmGbQuui75FiyjAX72HX15DwBBwF9dNiQZRQAg9nnPhYy+TUnE0+VcrttuvNI8oSxZcocA==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-core-module": "^2.13.0",

"path-parse": "^1.0.7",

"supports-preserve-symlinks-flag": "^1.0.0"

},

"bin": {

"resolve": "bin/resolve"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/resolve-from": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/resolve-from/-/resolve-from-4.0.0.tgz",

"integrity": "sha512-pb/MYmXstAkysRFx8piNI1tGFNQIFA3vkE3Gq4EuA1dF6gHp/+vgZqsCGJapvy8N3Q+4o7FwvquPJcnZ7RYy4g==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=4"

}

},

"node\_modules/reusify": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/reusify/-/reusify-1.0.4.tgz",

"integrity": "sha512-U9nH88a3fc/ekCF1l0/UP1IosiuIjyTh7hBvXVMHYgVcfGvt897Xguj2UOLDeI5BG2m7/uwyaLVT6fbtCwTyzw==",

"dev": true,

"license": "MIT",

"engines": {

"iojs": ">=1.0.0",

"node": ">=0.10.0"

}

},

"node\_modules/rimraf": {

"version": "3.0.2",

"resolved": "https://registry.npmjs.org/rimraf/-/rimraf-3.0.2.tgz",

"integrity": "sha512-JZkJMZkAGFFPP2YqXZXPbMlMBgsxzE8ILs4lMIX/2o0L9UBw9O/Y3o6wFw/i9YLapcUJWwqbi3kdxIPdC62TIA==",

"deprecated": "Rimraf versions prior to v4 are no longer supported",

"dev": true,

"license": "ISC",

"dependencies": {

"glob": "^7.1.3"

},

"bin": {

"rimraf": "bin.js"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/rollup": {

"version": "4.20.0",

"resolved": "https://registry.npmjs.org/rollup/-/rollup-4.20.0.tgz",

"integrity": "sha512-6rbWBChcnSGzIlXeIdNIZTopKYad8ZG8ajhl78lGRLsI2rX8IkaotQhVas2Ma+GPxJav19wrSzvRvuiv0YKzWw==",

"dev": true,

"license": "MIT",

"dependencies": {

"@types/estree": "1.0.5"

},

"bin": {

"rollup": "dist/bin/rollup"

},

"engines": {

"node": ">=18.0.0",

"npm": ">=8.0.0"

},

"optionalDependencies": {

"@rollup/rollup-android-arm-eabi": "4.20.0",

"@rollup/rollup-android-arm64": "4.20.0",

"@rollup/rollup-darwin-arm64": "4.20.0",

"@rollup/rollup-darwin-x64": "4.20.0",

"@rollup/rollup-linux-arm-gnueabihf": "4.20.0",

"@rollup/rollup-linux-arm-musleabihf": "4.20.0",

"@rollup/rollup-linux-arm64-gnu": "4.20.0",

"@rollup/rollup-linux-arm64-musl": "4.20.0",

"@rollup/rollup-linux-powerpc64le-gnu": "4.20.0",

"@rollup/rollup-linux-riscv64-gnu": "4.20.0",

"@rollup/rollup-linux-s390x-gnu": "4.20.0",

"@rollup/rollup-linux-x64-gnu": "4.20.0",

"@rollup/rollup-linux-x64-musl": "4.20.0",

"@rollup/rollup-win32-arm64-msvc": "4.20.0",

"@rollup/rollup-win32-ia32-msvc": "4.20.0",

"@rollup/rollup-win32-x64-msvc": "4.20.0",

"fsevents": "~2.3.2"

}

},

"node\_modules/run-parallel": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/run-parallel/-/run-parallel-1.2.0.tgz",

"integrity": "sha512-5l4VyZR86LZ/lDxZTR6jqL8AFE2S0IFLMP26AbjsLVADxHdhB/c0GUsH+y39UfCi3dzz8OlQuPmnaJOMoDHQBA==",

"dev": true,

"funding": [

{

"type": "github",

"url": "https://github.com/sponsors/feross"

},

{

"type": "patreon",

"url": "https://www.patreon.com/feross"

},

{

"type": "consulting",

"url": "https://feross.org/support"

}

],

"license": "MIT",

"dependencies": {

"queue-microtask": "^1.2.2"

}

},

"node\_modules/safe-array-concat": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/safe-array-concat/-/safe-array-concat-1.1.2.tgz",

"integrity": "sha512-vj6RsCsWBCf19jIeHEfkRMw8DPiBb+DMXklQ/1SGDHOMlHdPUkZXFQ2YdplS23zESTijAcurb1aSgJA3AgMu1Q==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"get-intrinsic": "^1.2.4",

"has-symbols": "^1.0.3",

"isarray": "^2.0.5"

},

"engines": {

"node": ">=0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/safe-regex-test": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/safe-regex-test/-/safe-regex-test-1.0.3.tgz",

"integrity": "sha512-CdASjNJPvRa7roO6Ra/gLYBTzYzzPyyBXxIMdGW3USQLyjWEls2RgW5UBTXaQVp+OrpeCK3bLem8smtmheoRuw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.6",

"es-errors": "^1.3.0",

"is-regex": "^1.1.4"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/scheduler": {

"version": "0.23.2",

"resolved": "https://registry.npmjs.org/scheduler/-/scheduler-0.23.2.tgz",

"integrity": "sha512-UOShsPwz7NrMUqhR6t0hWjFduvOzbtv7toDH1/hIrfRNIDBnnBWd0CwJTGvTpngVlmwGCdP9/Zl/tVrDqcuYzQ==",

"license": "MIT",

"dependencies": {

"loose-envify": "^1.1.0"

}

},

"node\_modules/semver": {

"version": "6.3.1",

"resolved": "https://registry.npmjs.org/semver/-/semver-6.3.1.tgz",

"integrity": "sha512-BR7VvDCVHO+q2xBEWskxS6DJE1qRnb7DxzUrogb71CWoSficBxYsiAGd+Kl0mmq/MprG9yArRkyrQxTO6XjMzA==",

"dev": true,

"license": "ISC",

"bin": {

"semver": "bin/semver.js"

}

},

"node\_modules/set-function-length": {

"version": "1.2.2",

"resolved": "https://registry.npmjs.org/set-function-length/-/set-function-length-1.2.2.tgz",

"integrity": "sha512-pgRc4hJ4/sNjWCSS9AmnS40x3bNMDTknHgL5UaMBTMyJnU90EgWh1Rz+MC9eFu4BuN/UwZjKQuY/1v3rM7HMfg==",

"dev": true,

"license": "MIT",

"dependencies": {

"define-data-property": "^1.1.4",

"es-errors": "^1.3.0",

"function-bind": "^1.1.2",

"get-intrinsic": "^1.2.4",

"gopd": "^1.0.1",

"has-property-descriptors": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/set-function-name": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/set-function-name/-/set-function-name-2.0.2.tgz",

"integrity": "sha512-7PGFlmtwsEADb0WYyvCMa1t+yke6daIG4Wirafur5kcf+MhUnPms1UeR0CKQdTZD81yESwMHbtn+TR+dMviakQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"define-data-property": "^1.1.4",

"es-errors": "^1.3.0",

"functions-have-names": "^1.2.3",

"has-property-descriptors": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/shebang-command": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/shebang-command/-/shebang-command-2.0.0.tgz",

"integrity": "sha512-kHxr2zZpYtdmrN1qDjrrX/Z1rR1kG8Dx+gkpK1G4eXmvXswmcE1hTWBWYUzlraYw1/yZp6YuDY77YtvbN0dmDA==",

"dev": true,

"license": "MIT",

"dependencies": {

"shebang-regex": "^3.0.0"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/shebang-regex": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/shebang-regex/-/shebang-regex-3.0.0.tgz",

"integrity": "sha512-7++dFhtcx3353uBaq8DDR4NuxBetBzC7ZQOhmTQInHEd6bSrXdiEyzCvG07Z44UYdLShWUyXt5M/yhz8ekcb1A==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/side-channel": {

"version": "1.0.6",

"resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.0.6.tgz",

"integrity": "sha512-fDW/EZ6Q9RiO8eFG8Hj+7u/oW+XrPTIChwCOM2+th2A6OblDtYYIpve9m+KvI9Z4C9qSEXlaGR6bTEYHReuglA==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.4",

"object-inspect": "^1.13.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/signal-exit": {

"version": "4.1.0",

"resolved": "https://registry.npmjs.org/signal-exit/-/signal-exit-4.1.0.tgz",

"integrity": "sha512-bzyZ1e88w9O1iNJbKnOlvYTrWPDl46O1bG0D3XInv+9tkPrxrN8jUUTiFlDkkmKWgn1M6CfIA13SuGqOa9Korw==",

"dev": true,

"license": "ISC",

"engines": {

"node": ">=14"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/source-map-js": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/source-map-js/-/source-map-js-1.2.0.tgz",

"integrity": "sha512-itJW8lvSA0TXEphiRoawsCksnlf8SyvmFzIhltqAHluXd88pkCd+cXJVHTDwdCr0IzwptSm035IHQktUu1QUMg==",

"dev": true,

"license": "BSD-3-Clause",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/string-width": {

"version": "5.1.2",

"resolved": "https://registry.npmjs.org/string-width/-/string-width-5.1.2.tgz",

"integrity": "sha512-HnLOCR3vjcY8beoNLtcjZ5/nxn2afmME6lhrDrebokqMap+XbeW8n9TXpPDOqdGK5qcI3oT0GKTW6wC7EMiVqA==",

"dev": true,

"license": "MIT",

"dependencies": {

"eastasianwidth": "^0.2.0",

"emoji-regex": "^9.2.2",

"strip-ansi": "^7.0.1"

},

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/string-width-cjs": {

"name": "string-width",

"version": "4.2.3",

"resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",

"integrity": "sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWyWUwvSTriiZz/g==",

"dev": true,

"license": "MIT",

"dependencies": {

"emoji-regex": "^8.0.0",

"is-fullwidth-code-point": "^3.0.0",

"strip-ansi": "^6.0.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/string-width-cjs/node\_modules/emoji-regex": {

"version": "8.0.0",

"resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",

"integrity": "sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==",

"dev": true,

"license": "MIT"

},

"node\_modules/string-width/node\_modules/ansi-regex": {

"version": "6.0.1",

"resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-6.0.1.tgz",

"integrity": "sha512-n5M855fKb2SsfMIiFFoVrABHJC8QtHwVx+mHWP3QcEqBHYienj5dHSgjbxtC0WEZXYt4wcD6zrQElDPhFuZgfA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/ansi-regex?sponsor=1"

}

},

"node\_modules/string-width/node\_modules/strip-ansi": {

"version": "7.1.0",

"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-7.1.0.tgz",

"integrity": "sha512-iq6eVVI64nQQTRYq2KtEg2d2uU7LElhTJwsH4YzIHZshxlgZms/wIc4VoDQTlG/IvVIrBKG06CrZnp0qv7hkcQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-regex": "^6.0.1"

},

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/strip-ansi?sponsor=1"

}

},

"node\_modules/string.prototype.matchall": {

"version": "4.0.11",

"resolved": "https://registry.npmjs.org/string.prototype.matchall/-/string.prototype.matchall-4.0.11.tgz",

"integrity": "sha512-NUdh0aDavY2og7IbBPenWqR9exH+E26Sv8e0/eTe1tltDGZL+GtBkDAnnyBtmekfK6/Dq3MkcGtzXFEd1LQrtg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.2",

"es-errors": "^1.3.0",

"es-object-atoms": "^1.0.0",

"get-intrinsic": "^1.2.4",

"gopd": "^1.0.1",

"has-symbols": "^1.0.3",

"internal-slot": "^1.0.7",

"regexp.prototype.flags": "^1.5.2",

"set-function-name": "^2.0.2",

"side-channel": "^1.0.6"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/string.prototype.repeat": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/string.prototype.repeat/-/string.prototype.repeat-1.0.0.tgz",

"integrity": "sha512-0u/TldDbKD8bFCQ/4f5+mNRrXwZ8hg2w7ZR8wa16e8z9XpePWl3eGEcUD0OXpEH/VJH/2G3gjUtR3ZOiBe2S/w==",

"dev": true,

"license": "MIT",

"dependencies": {

"define-properties": "^1.1.3",

"es-abstract": "^1.17.5"

}

},

"node\_modules/string.prototype.trim": {

"version": "1.2.9",

"resolved": "https://registry.npmjs.org/string.prototype.trim/-/string.prototype.trim-1.2.9.tgz",

"integrity": "sha512-klHuCNxiMZ8MlsOihJhJEBJAiMVqU3Z2nEXWfWnIqjN0gEFS9J9+IxKozWWtQGcgoa1WUZzLjKPTr4ZHNFTFxw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-abstract": "^1.23.0",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/string.prototype.trimend": {

"version": "1.0.8",

"resolved": "https://registry.npmjs.org/string.prototype.trimend/-/string.prototype.trimend-1.0.8.tgz",

"integrity": "sha512-p73uL5VCHCO2BZZ6krwwQE3kCzM7NKmis8S//xEC6fQonchbum4eP6kR4DLEjQFO3Wnj3Fuo8NM0kOSjVdHjZQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-object-atoms": "^1.0.0"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/string.prototype.trimstart": {

"version": "1.0.8",

"resolved": "https://registry.npmjs.org/string.prototype.trimstart/-/string.prototype.trimstart-1.0.8.tgz",

"integrity": "sha512-UXSH262CSZY1tfu3G3Secr6uGLCFVPMhIqHjlgCUtCCcgihYc/xKs9djMTMUOb2j1mVSeU8EU6NWc/iQKU6Gfg==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"define-properties": "^1.2.1",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/strip-ansi": {

"version": "6.0.1",

"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",

"integrity": "sha512-Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ooI7KrPuUSztUdU5A==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-regex": "^5.0.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/strip-ansi-cjs": {

"name": "strip-ansi",

"version": "6.0.1",

"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",

"integrity": "sha512-Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ooI7KrPuUSztUdU5A==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-regex": "^5.0.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/strip-json-comments": {

"version": "3.1.1",

"resolved": "https://registry.npmjs.org/strip-json-comments/-/strip-json-comments-3.1.1.tgz",

"integrity": "sha512-6fPc+R4ihwqP6N/aIv2f1gMH8lOVtWQHoqC4yK6oSDVVocumAsfCqjkXnqiYMhmMwS/mEHLp7Vehlt3ql6lEig==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/sucrase": {

"version": "3.35.0",

"resolved": "https://registry.npmjs.org/sucrase/-/sucrase-3.35.0.tgz",

"integrity": "sha512-8EbVDiu9iN/nESwxeSxDKe0dunta1GOlHufmSSXxMD2z2/tMZpDMpvXQGsc+ajGo8y2uYUmixaSRUc/QPoQ0GA==",

"dev": true,

"license": "MIT",

"dependencies": {

"@jridgewell/gen-mapping": "^0.3.2",

"commander": "^4.0.0",

"glob": "^10.3.10",

"lines-and-columns": "^1.1.6",

"mz": "^2.7.0",

"pirates": "^4.0.1",

"ts-interface-checker": "^0.1.9"

},

"bin": {

"sucrase": "bin/sucrase",

"sucrase-node": "bin/sucrase-node"

},

"engines": {

"node": ">=16 || 14 >=14.17"

}

},

"node\_modules/sucrase/node\_modules/brace-expansion": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.1.tgz",

"integrity": "sha512-XnAIvQ8eM+kC6aULx6wuQiwVsnzsi9d3WxzV3FpWTGA19F621kwdbsAcFKXgKUHZWsy+mY6iL1sHTxWEFCytDA==",

"dev": true,

"license": "MIT",

"dependencies": {

"balanced-match": "^1.0.0"

}

},

"node\_modules/sucrase/node\_modules/glob": {

"version": "10.4.5",

"resolved": "https://registry.npmjs.org/glob/-/glob-10.4.5.tgz",

"integrity": "sha512-7Bv8RF0k6xjo7d4A/PxYLbUCfb6c+Vpd2/mB2yRDlew7Jb5hEXiCD9ibfO7wpk8i4sevK6DFny9h7EYbM3/sHg==",

"dev": true,

"license": "ISC",

"dependencies": {

"foreground-child": "^3.1.0",

"jackspeak": "^3.1.2",

"minimatch": "^9.0.4",

"minipass": "^7.1.2",

"package-json-from-dist": "^1.0.0",

"path-scurry": "^1.11.1"

},

"bin": {

"glob": "dist/esm/bin.mjs"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/sucrase/node\_modules/minimatch": {

"version": "9.0.5",

"resolved": "https://registry.npmjs.org/minimatch/-/minimatch-9.0.5.tgz",

"integrity": "sha512-G6T0ZX48xgozx7587koeX9Ys2NYy6Gmv//P89sEte9V9whIapMNF4idKxnW2QtCcLiTWlb/wfCabAtAFWhhBow==",

"dev": true,

"license": "ISC",

"dependencies": {

"brace-expansion": "^2.0.1"

},

"engines": {

"node": ">=16 || 14 >=14.17"

},

"funding": {

"url": "https://github.com/sponsors/isaacs"

}

},

"node\_modules/supports-color": {

"version": "5.5.0",

"resolved": "https://registry.npmjs.org/supports-color/-/supports-color-5.5.0.tgz",

"integrity": "sha512-QjVjwdXIt408MIiAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFNTr4uMNZma0ey4f5lgLrkB0aX0QMow==",

"dev": true,

"license": "MIT",

"dependencies": {

"has-flag": "^3.0.0"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/supports-preserve-symlinks-flag": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/supports-preserve-symlinks-flag/-/supports-preserve-symlinks-flag-1.0.0.tgz",

"integrity": "sha512-ot0WnXS9fgdkgIcePe6RHNk1WA8+muPa6cSjeR3V8K27q9BB1rTE3R1p7Hv0z1ZyAc8s6Vvv8DIyWf681MAt0w==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/tailwindcss": {

"version": "3.4.9",

"resolved": "https://registry.npmjs.org/tailwindcss/-/tailwindcss-3.4.9.tgz",

"integrity": "sha512-1SEOvRr6sSdV5IDf9iC+NU4dhwdqzF4zKKq3sAbasUWHEM6lsMhX+eNN5gkPx1BvLFEnZQEUFbXnGj8Qlp83Pg==",

"dev": true,

"license": "MIT",

"dependencies": {

"@alloc/quick-lru": "^5.2.0",

"arg": "^5.0.2",

"chokidar": "^3.5.3",

"didyoumean": "^1.2.2",

"dlv": "^1.1.3",

"fast-glob": "^3.3.0",

"glob-parent": "^6.0.2",

"is-glob": "^4.0.3",

"jiti": "^1.21.0",

"lilconfig": "^2.1.0",

"micromatch": "^4.0.5",

"normalize-path": "^3.0.0",

"object-hash": "^3.0.0",

"picocolors": "^1.0.0",

"postcss": "^8.4.23",

"postcss-import": "^15.1.0",

"postcss-js": "^4.0.1",

"postcss-load-config": "^4.0.1",

"postcss-nested": "^6.0.1",

"postcss-selector-parser": "^6.0.11",

"resolve": "^1.22.2",

"sucrase": "^3.32.0"

},

"bin": {

"tailwind": "lib/cli.js",

"tailwindcss": "lib/cli.js"

},

"engines": {

"node": ">=14.0.0"

}

},

"node\_modules/tailwindcss/node\_modules/resolve": {

"version": "1.22.8",

"resolved": "https://registry.npmjs.org/resolve/-/resolve-1.22.8.tgz",

"integrity": "sha512-oKWePCxqpd6FlLvGV1VU0x7bkPmmCNolxzjMf4NczoDnQcIWrAF+cPtZn5i6n+RfD2d9i0tzpKnG6Yk168yIyw==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-core-module": "^2.13.0",

"path-parse": "^1.0.7",

"supports-preserve-symlinks-flag": "^1.0.0"

},

"bin": {

"resolve": "bin/resolve"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/text-table": {

"version": "0.2.0",

"resolved": "https://registry.npmjs.org/text-table/-/text-table-0.2.0.tgz",

"integrity": "sha512-N+8UisAXDGk8PFXP4HAzVR9nbfmVJ3zYLAWiTIoqC5v5isinhr+r5uaO8+7r3BMfuNIufIsA7RdpVgacC2cSpw==",

"dev": true,

"license": "MIT"

},

"node\_modules/thenify": {

"version": "3.3.1",

"resolved": "https://registry.npmjs.org/thenify/-/thenify-3.3.1.tgz",

"integrity": "sha512-RVZSIV5IG10Hk3enotrhvz0T9em6cyHBLkH/YAZuKqd8hRkKhSfCGIcP2KUY0EPxndzANBmNllzWPwak+bheSw==",

"dev": true,

"license": "MIT",

"dependencies": {

"any-promise": "^1.0.0"

}

},

"node\_modules/thenify-all": {

"version": "1.6.0",

"resolved": "https://registry.npmjs.org/thenify-all/-/thenify-all-1.6.0.tgz",

"integrity": "sha512-RNxQH/qI8/t3thXJDwcstUO4zeqo64+Uy/+sNVRBx4Xn2OX+OZ9oP+iJnNFqplFra2ZUVeKCSa2oVWi3T4uVmA==",

"dev": true,

"license": "MIT",

"dependencies": {

"thenify": ">= 3.1.0 < 4"

},

"engines": {

"node": ">=0.8"

}

},

"node\_modules/to-fast-properties": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/to-fast-properties/-/to-fast-properties-2.0.0.tgz",

"integrity": "sha512-/OaKK0xYrs3DmxRYqL/yDc+FxFUVYhDlXMhRmv3z915w2HF1tnN1omB354j8VUGO/hbRzyD6Y3sA7v7GS/ceog==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=4"

}

},

"node\_modules/to-regex-range": {

"version": "5.0.1",

"resolved": "https://registry.npmjs.org/to-regex-range/-/to-regex-range-5.0.1.tgz",

"integrity": "sha512-65P7iz6X5yEr1cwcgvQxbbIw7Uk3gOy5dIdtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-number": "^7.0.0"

},

"engines": {

"node": ">=8.0"

}

},

"node\_modules/ts-interface-checker": {

"version": "0.1.13",

"resolved": "https://registry.npmjs.org/ts-interface-checker/-/ts-interface-checker-0.1.13.tgz",

"integrity": "sha512-Y/arvbn+rrz3JCKl9C4kVNfTfSm2/mEp5FSz5EsZSANGPSlQrpRI5M4PKF+mJnE52jOO90PnPSc3Ur3bTQw0gA==",

"dev": true,

"license": "Apache-2.0"

},

"node\_modules/type-check": {

"version": "0.4.0",

"resolved": "https://registry.npmjs.org/type-check/-/type-check-0.4.0.tgz",

"integrity": "sha512-XleUoc9uwGXqjWwXaUTZAmzMcFZ5858QA2vvx1Ur5xIcixXIP+8LnFDgRplU30us6teqdlskFfu+ae4K79Ooew==",

"dev": true,

"license": "MIT",

"dependencies": {

"prelude-ls": "^1.2.1"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/type-fest": {

"version": "0.20.2",

"resolved": "https://registry.npmjs.org/type-fest/-/type-fest-0.20.2.tgz",

"integrity": "sha512-Ne+eE4r0/iWnpAxD852z3A+N0Bt5RN//NjJwRd2VFHEmrywxf5vsZlh4R6lixl6B+wz/8d+maTSAkN1FIkI3LQ==",

"dev": true,

"license": "(MIT OR CC0-1.0)",

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/typed-array-buffer": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/typed-array-buffer/-/typed-array-buffer-1.0.2.tgz",

"integrity": "sha512-gEymJYKZtKXzzBzM4jqa9w6Q1Jjm7x2d+sh19AdsD4wqnMPDYyvwpsIc2Q/835kHuo3BEQ7CjelGhfTsoBb2MQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"es-errors": "^1.3.0",

"is-typed-array": "^1.1.13"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/typed-array-byte-length": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/typed-array-byte-length/-/typed-array-byte-length-1.0.1.tgz",

"integrity": "sha512-3iMJ9q0ao7WE9tWcaYKIptkNBuOIcZCCT0d4MRvuuH88fEoEH62IuQe0OtraD3ebQEoTRk8XCBoknUNc1Y67pw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"for-each": "^0.3.3",

"gopd": "^1.0.1",

"has-proto": "^1.0.3",

"is-typed-array": "^1.1.13"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/typed-array-byte-offset": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/typed-array-byte-offset/-/typed-array-byte-offset-1.0.2.tgz",

"integrity": "sha512-Ous0vodHa56FviZucS2E63zkgtgrACj7omjwd/8lTEMEPFFyjfixMZ1ZXenpgCFBBt4EC1J2XsyVS2gkG0eTFA==",

"dev": true,

"license": "MIT",

"dependencies": {

"available-typed-arrays": "^1.0.7",

"call-bind": "^1.0.7",

"for-each": "^0.3.3",

"gopd": "^1.0.1",

"has-proto": "^1.0.3",

"is-typed-array": "^1.1.13"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/typed-array-length": {

"version": "1.0.6",

"resolved": "https://registry.npmjs.org/typed-array-length/-/typed-array-length-1.0.6.tgz",

"integrity": "sha512-/OxDN6OtAk5KBpGb28T+HZc2M+ADtvRxXrKKbUwtsLgdoxgX13hyy7ek6bFRl5+aBs2yZzB0c4CnQfAtVypW/g==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.7",

"for-each": "^0.3.3",

"gopd": "^1.0.1",

"has-proto": "^1.0.3",

"is-typed-array": "^1.1.13",

"possible-typed-array-names": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/unbox-primitive": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/unbox-primitive/-/unbox-primitive-1.0.2.tgz",

"integrity": "sha512-61pPlCD9h51VoreyJ0BReideM3MDKMKnh6+V9L08331ipq6Q8OFXZYiqP6n/tbHx4s5I9uRhcye6BrbkizkBDw==",

"dev": true,

"license": "MIT",

"dependencies": {

"call-bind": "^1.0.2",

"has-bigints": "^1.0.2",

"has-symbols": "^1.0.3",

"which-boxed-primitive": "^1.0.2"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/update-browserslist-db": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/update-browserslist-db/-/update-browserslist-db-1.1.0.tgz",

"integrity": "sha512-EdRAaAyk2cUE1wOf2DkEhzxqOQvFOoRJFNS6NeyJ01Gp2beMRpBAINjM2iDXE3KCuKhwnvHIQCJm6ThL2Z+HzQ==",

"dev": true,

"funding": [

{

"type": "opencollective",

"url": "https://opencollective.com/browserslist"

},

{

"type": "tidelift",

"url": "https://tidelift.com/funding/github/npm/browserslist"

},

{

"type": "github",

"url": "https://github.com/sponsors/ai"

}

],

"license": "MIT",

"dependencies": {

"escalade": "^3.1.2",

"picocolors": "^1.0.1"

},

"bin": {

"update-browserslist-db": "cli.js"

},

"peerDependencies": {

"browserslist": ">= 4.21.0"

}

},

"node\_modules/uri-js": {

"version": "4.4.1",

"resolved": "https://registry.npmjs.org/uri-js/-/uri-js-4.4.1.tgz",

"integrity": "sha512-7rKUyy33Q1yc98pQ1DAmLtwX109F7TIfWlW1Ydo8Wl1ii1SeHieeh0HHfPeL2fMXK6z0s8ecKs9frCuLJvndBg==",

"dev": true,

"license": "BSD-2-Clause",

"dependencies": {

"punycode": "^2.1.0"

}

},

"node\_modules/util-deprecate": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",

"integrity": "sha512-EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAIiI2TaSCA7VCJ1UJDMagCzIkXNsUYfD1daK//LTEQ8xiIbrHtcw==",

"dev": true,

"license": "MIT"

},

"node\_modules/vite": {

"version": "5.4.0",

"resolved": "https://registry.npmjs.org/vite/-/vite-5.4.0.tgz",

"integrity": "sha512-5xokfMX0PIiwCMCMb9ZJcMyh5wbBun0zUzKib+L65vAZ8GY9ePZMXxFrHbr/Kyll2+LSCY7xtERPpxkBDKngwg==",

"dev": true,

"license": "MIT",

"dependencies": {

"esbuild": "^0.21.3",

"postcss": "^8.4.40",

"rollup": "^4.13.0"

},

"bin": {

"vite": "bin/vite.js"

},

"engines": {

"node": "^18.0.0 || >=20.0.0"

},

"funding": {

"url": "https://github.com/vitejs/vite?sponsor=1"

},

"optionalDependencies": {

"fsevents": "~2.3.3"

},

"peerDependencies": {

"@types/node": "^18.0.0 || >=20.0.0",

"less": "\*",

"lightningcss": "^1.21.0",

"sass": "\*",

"sass-embedded": "\*",

"stylus": "\*",

"sugarss": "\*",

"terser": "^5.4.0"

},

"peerDependenciesMeta": {

"@types/node": {

"optional": true

},

"less": {

"optional": true

},

"lightningcss": {

"optional": true

},

"sass": {

"optional": true

},

"sass-embedded": {

"optional": true

},

"stylus": {

"optional": true

},

"sugarss": {

"optional": true

},

"terser": {

"optional": true

}

}

},

"node\_modules/which": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/which/-/which-2.0.2.tgz",

"integrity": "sha512-BLI3Tl1TW3Pvl70l3yq3Y64i+awpwXqsGBYWkkqMtnbXgrMD+yj7rhW0kuEDxzJaYXGjEW5ogapKNMEKNMjibA==",

"dev": true,

"license": "ISC",

"dependencies": {

"isexe": "^2.0.0"

},

"bin": {

"node-which": "bin/node-which"

},

"engines": {

"node": ">= 8"

}

},

"node\_modules/which-boxed-primitive": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/which-boxed-primitive/-/which-boxed-primitive-1.0.2.tgz",

"integrity": "sha512-bwZdv0AKLpplFY2KZRX6TvyuN7ojjr7lwkg6ml0roIy9YeuSr7JS372qlNW18UQYzgYK9ziGcerWqZOmEn9VNg==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-bigint": "^1.0.1",

"is-boolean-object": "^1.1.0",

"is-number-object": "^1.0.4",

"is-string": "^1.0.5",

"is-symbol": "^1.0.3"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/which-builtin-type": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/which-builtin-type/-/which-builtin-type-1.1.4.tgz",

"integrity": "sha512-bppkmBSsHFmIMSl8BO9TbsyzsvGjVoppt8xUiGzwiu/bhDCGxnpOKCxgqj6GuyHE0mINMDecBFPlOm2hzY084w==",

"dev": true,

"license": "MIT",

"dependencies": {

"function.prototype.name": "^1.1.6",

"has-tostringtag": "^1.0.2",

"is-async-function": "^2.0.0",

"is-date-object": "^1.0.5",

"is-finalizationregistry": "^1.0.2",

"is-generator-function": "^1.0.10",

"is-regex": "^1.1.4",

"is-weakref": "^1.0.2",

"isarray": "^2.0.5",

"which-boxed-primitive": "^1.0.2",

"which-collection": "^1.0.2",

"which-typed-array": "^1.1.15"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/which-collection": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/which-collection/-/which-collection-1.0.2.tgz",

"integrity": "sha512-K4jVyjnBdgvc86Y6BkaLZEN933SwYOuBFkdmBu9ZfkcAbdVbpITnDmjvZ/aQjRXQrv5EPkTnD1s39GiiqbngCw==",

"dev": true,

"license": "MIT",

"dependencies": {

"is-map": "^2.0.3",

"is-set": "^2.0.3",

"is-weakmap": "^2.0.2",

"is-weakset": "^2.0.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/which-typed-array": {

"version": "1.1.15",

"resolved": "https://registry.npmjs.org/which-typed-array/-/which-typed-array-1.1.15.tgz",

"integrity": "sha512-oV0jmFtUky6CXfkqehVvBP/LSWJ2sy4vWMioiENyJLePrBO/yKyV9OyJySfAKosh+RYkIl5zJCNZ8/4JncrpdA==",

"dev": true,

"license": "MIT",

"dependencies": {

"available-typed-arrays": "^1.0.7",

"call-bind": "^1.0.7",

"for-each": "^0.3.3",

"gopd": "^1.0.1",

"has-tostringtag": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/word-wrap": {

"version": "1.2.5",

"resolved": "https://registry.npmjs.org/word-wrap/-/word-wrap-1.2.5.tgz",

"integrity": "sha512-BN22B5eaMMI9UMtjrGd5g5eCYPpCPDUy0FJXbYsaT5zYxjFOckS53SQDE3pWkVoWpHXVb3BrYcEN4Twa55B5cA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/wrap-ansi": {

"version": "8.1.0",

"resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-8.1.0.tgz",

"integrity": "sha512-si7QWI6zUMq56bESFvagtmzMdGOtoxfR+Sez11Mobfc7tm+VkUckk9bW2UeffTGVUbOksxmSw0AA2gs8g71NCQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-styles": "^6.1.0",

"string-width": "^5.0.1",

"strip-ansi": "^7.0.1"

},

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/wrap-ansi?sponsor=1"

}

},

"node\_modules/wrap-ansi-cjs": {

"name": "wrap-ansi",

"version": "7.0.0",

"resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-7.0.0.tgz",

"integrity": "sha512-YVGIj2kamLSTxw6NsZjoBxfSwsn0ycdesmc4p+Q21c5zPuZ1pl+NfxVdxPtdHvmNVOQ6XSYG4AUtyt/Fi7D16Q==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-styles": "^4.0.0",

"string-width": "^4.1.0",

"strip-ansi": "^6.0.0"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/chalk/wrap-ansi?sponsor=1"

}

},

"node\_modules/wrap-ansi-cjs/node\_modules/ansi-styles": {

"version": "4.3.0",

"resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",

"integrity": "sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPlFp222wVALIheZJQSEg==",

"dev": true,

"license": "MIT",

"dependencies": {

"color-convert": "^2.0.1"

},

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/chalk/ansi-styles?sponsor=1"

}

},

"node\_modules/wrap-ansi-cjs/node\_modules/color-convert": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",

"integrity": "sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHCaElKQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"color-name": "~1.1.4"

},

"engines": {

"node": ">=7.0.0"

}

},

"node\_modules/wrap-ansi-cjs/node\_modules/color-name": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",

"integrity": "sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGXgAsQA==",

"dev": true,

"license": "MIT"

},

"node\_modules/wrap-ansi-cjs/node\_modules/emoji-regex": {

"version": "8.0.0",

"resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",

"integrity": "sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==",

"dev": true,

"license": "MIT"

},

"node\_modules/wrap-ansi-cjs/node\_modules/string-width": {

"version": "4.2.3",

"resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",

"integrity": "sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWyWUwvSTriiZz/g==",

"dev": true,

"license": "MIT",

"dependencies": {

"emoji-regex": "^8.0.0",

"is-fullwidth-code-point": "^3.0.0",

"strip-ansi": "^6.0.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/wrap-ansi/node\_modules/ansi-regex": {

"version": "6.0.1",

"resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-6.0.1.tgz",

"integrity": "sha512-n5M855fKb2SsfMIiFFoVrABHJC8QtHwVx+mHWP3QcEqBHYienj5dHSgjbxtC0WEZXYt4wcD6zrQElDPhFuZgfA==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/ansi-regex?sponsor=1"

}

},

"node\_modules/wrap-ansi/node\_modules/ansi-styles": {

"version": "6.2.1",

"resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-6.2.1.tgz",

"integrity": "sha512-bN798gFfQX+viw3R7yrGWRqnrN2oRkEkUjjl4JNn4E8GxxbjtG3FbrEIIY3l8/hrwUwIeCZvi4QuOTP4MErVug==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/ansi-styles?sponsor=1"

}

},

"node\_modules/wrap-ansi/node\_modules/strip-ansi": {

"version": "7.1.0",

"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-7.1.0.tgz",

"integrity": "sha512-iq6eVVI64nQQTRYq2KtEg2d2uU7LElhTJwsH4YzIHZshxlgZms/wIc4VoDQTlG/IvVIrBKG06CrZnp0qv7hkcQ==",

"dev": true,

"license": "MIT",

"dependencies": {

"ansi-regex": "^6.0.1"

},

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://github.com/chalk/strip-ansi?sponsor=1"

}

},

"node\_modules/wrappy": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",

"integrity": "sha512-l4Sp/DRseor9wL6EvV2+TuQn63dMkPjZ/sp9XkghTEbV9KlPS1xUsZ3u7/IQO4wxtcFB4bgpQPRcR3QCvezPcQ==",

"dev": true,

"license": "ISC"

},

"node\_modules/yallist": {

"version": "3.1.1",

"resolved": "https://registry.npmjs.org/yallist/-/yallist-3.1.1.tgz",

"integrity": "sha512-a4UGQaWPH59mOXUYnAG2ewncQS4i4F43Tv3JoAM+s2VDAmS9NsK8GpDMLrCHPksFT7h3K6TOoUNn2pb7RoXx4g==",

"dev": true,

"license": "ISC"

},

"node\_modules/yaml": {

"version": "2.5.0",

"resolved": "https://registry.npmjs.org/yaml/-/yaml-2.5.0.tgz",

"integrity": "sha512-2wWLbGbYDiSqqIKoPjar3MPgB94ErzCtrNE1FdqGuaO0pi2JGjmE8aW8TDZwzU7vuxcGRdL/4gPQwQ7hD5AMSw==",

"dev": true,

"license": "ISC",

"bin": {

"yaml": "bin.mjs"

},

"engines": {

"node": ">= 14"

}

},

"node\_modules/yocto-queue": {

"version": "0.1.0",

"resolved": "https://registry.npmjs.org/yocto-queue/-/yocto-queue-0.1.0.tgz",

"integrity": "sha512-rVksvsnNCdJ/ohGc6xgPwyN8eheCxsiLM8mxuE/t/mOVqJewPuO1miLpTHQiRgTKCLexL4MeAFVagts7HmNZ2Q==",

"dev": true,

"license": "MIT",

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

}

}

}

**package.json:**

{

  "name": "frontend",

  "private": true,

  "version": "0.0.0",

  "type": "module",

  "scripts": {

    "dev": "vite",

    "build": "vite build",

    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",

    "preview": "vite preview"

  },

  "dependencies": {

    "axios": "^1.7.3",

    "react": "^18.3.1",

    "react-dom": "^18.3.1",

    "react-router-dom": "^6.26.0",

    "react-toastify": "^10.0.5"

  },

  "devDependencies": {

    "@types/react": "^18.3.3",

    "@types/react-dom": "^18.3.0",

    "@vitejs/plugin-react": "^4.3.1",

    "autoprefixer": "^10.4.20",

    "eslint": "^8.57.0",

    "eslint-plugin-react": "^7.35.0",

    "eslint-plugin-react-hooks": "^4.6.2",

    "eslint-plugin-react-refresh": "^0.4.9",

    "postcss": "^8.4.41",

    "tailwindcss": "^3.4.9",

    "vite": "^5.4.0"

  }

}

**tailwind.config.js:**

/\*\* @type {import('tailwindcss').Config} \*/

export default {

  content: [

    "./index.html",

    "./src/\*\*/\*.{js,ts,jsx,tsx}",

  ],

  theme: {

    extend: {

      gridTemplateColumns:{

        'auto':'repeat(auto-fill, minmax(200px, 1fr))'

      },

      colors:{

        'primary':'#5F6FFF'

      }

    },

  },

  plugins: [],

}

**vercel.json:**

{

  "rewrites": [

    {

      "source": "/(.\*)",

      "destination": "/"

    }

  ]

}

**vite.config.js:**

import { defineConfig } from 'vite';

import react from '@vitejs/plugin-react';

export default defineConfig({

  plugins: [react()],

  server: {

    port: 5173,

    open: false,

    proxy: {

      '/api': {

        target: 'http://127.0.0.1:5002',

        changeOrigin: true,

      },

    },

  },

});

**Chatbot:**

**.env:**

API\_KEY=AIzaSyAxt1oBL387GkWkLgpswy6iMINrW4u1xpE

**.gitignore:**

.env

node\_modules/

**index.html:**

<!DOCTYPE html>

<html>

<head>

<title>Chatbot</title>

<style>

  body {

    font-family: sans-serif;

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

    background-color: #f0f0f0;

  }

  #chat-container {

    background-color: #fff;

    border-radius: 10px;

    padding: 20px;

    box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.1);

    width: 400px;

  }

  #chat-container p{

    text-align: center;

  }

  h1 {

    text-align: center;

    margin: 0;

  }

  #chat-history {

    height: 300px;

    overflow-y: scroll;

  }

  .user-message {

    text-align: right;

    padding: 10px;

    background-color: #f0f0f0;

    border-radius: 10px;

    margin-bottom: 5px;

  }

  .bot-message {

    text-align: left;

    padding: 10px;

    background-color: #e0f0e0;

    border-radius: 10px;

    margin-bottom: 5px;

  }

  form {

    display: flex;

    margin-top: 15px;

  }

  input {

    flex-grow: 1;

    margin-right: 10px;

    padding: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;

  }

  button {

    background-color: #4CAF50;

    color: white;

    border: none;

    padding: 10px 15px;

    border-radius: 5px;

    cursor: pointer;

  }

  #loader {

  display: none; /\* Hide by default \*/

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

}

  </style>

</head>

<body>

  <div id="chat-container">

    <h1>Dental Assistant</h1>

    <p>Your Dental Assistant!</p>

    <div id="chat-history"></div>

    <form id="chat-form">

      <input type="text" id="user-input" placeholder="Enter your message">

      <button type="submit">Send</button>

    </form>

</div>

<div id="loader">

  <img src="loader.gif" width="150px" alt="Loading...">

</div>

  <script>

    const chatHistory = document.getElementById('chat-history');

    const userInput = document.getElementById('user-input');

    const form = document.getElementById('chat-form');

    async function sendMessage() {

      const userMessage = userInput.value;

      userInput.value = ''; // Clear input field

      console.log(userMessage)

      try {

        const response = await fetch('/chat', {

          method: 'POST',

          headers: {

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

          },

          body: JSON.stringify({ userInput: userMessage }),

        });

        const data = await response.json();

        console.log(data)

        const botMessage = data.response;

        console.log(botMessage)

        // Add chat message to the chat history

        chatHistory.innerHTML += `<div class="user-message">${userMessage}</div>`;

        chatHistory.innerHTML += `<div class="bot-message">${botMessage}</div>`;

        // Scroll to the bottom of the chat history

        chatHistory.scrollTop = chatHistory.scrollHeight;

      } catch (error) {

        console.error('Error:', error);

        // Handle errors gracefully, e.g., display an error message to the user

      }

    }

    form.addEventListener('submit', (event) => {

      event.preventDefault(); // Prevent form submission

      const loader = document.getElementById('loader');

      loader.style.display = 'block'; // Show the loader

      sendMessage().finally(() => {

      loader.style.display = 'none'; // Hide the loader after the message is sent

    });;

    });

  </script>

</body>

</html>

**package-lock-json:**

{

  "name": "gemini\_chatbot",

  "version": "1.0.0",

  "lockfileVersion": 3,

  "requires": true,

  "packages": {

    "": {

      "name": "gemini\_chatbot",

      "version": "1.0.0",

      "license": "ISC",

      "dependencies": {

        "@google/generative-ai": "^0.1.3",

        "dotenv": "^16.3.1",

        "express": "^4.21.1"

      }

    },

    "node\_modules/@google/generative-ai": {

      "version": "0.1.3",

      "resolved": "https://registry.npmjs.org/@google/generative-ai/-/generative-ai-0.1.3.tgz",

      "integrity": "sha512-Cm4uJX1sKarpm1mje/MiOIinM7zdUUrQp/5/qGPAgznbdd/B9zup5ehT6c1qGqycFcSopTA1J1HpqHS5kJR8hQ==",

      "engines": {

        "node": ">=18.0.0"

      }

    },

    "node\_modules/accepts": {

      "version": "1.3.8",

      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",

      "integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NHcVF7rONl6qcaxV3Uuemwawk+7+SJLw==",

      "dependencies": {

        "mime-types": "~2.1.34",

        "negotiator": "0.6.3"

      },

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/array-flatten": {

      "version": "1.1.1",

      "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",

      "integrity": "sha512-PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNlM++NqRcK6CxxpUafjmhIdKiHibqg=="

    },

    "node\_modules/body-parser": {

      "version": "1.20.3",

      "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.20.3.tgz",

      "integrity": "sha512-7rAxByjUMqQ3/bHJy7D6OGXvx/MMc4IqBn/X0fcM1QUcAItpZrBEYhWGem+tzXH90c+G01ypMcYJBO9Y30203g==",

      "license": "MIT",

      "dependencies": {

        "bytes": "3.1.2",

        "content-type": "~1.0.5",

        "debug": "2.6.9",

        "depd": "2.0.0",

        "destroy": "1.2.0",

        "http-errors": "2.0.0",

        "iconv-lite": "0.4.24",

        "on-finished": "2.4.1",

        "qs": "6.13.0",

        "raw-body": "2.5.2",

        "type-is": "~1.6.18",

        "unpipe": "1.0.0"

      },

      "engines": {

        "node": ">= 0.8",

        "npm": "1.2.8000 || >= 1.4.16"

      }

    },

    "node\_modules/bytes": {

      "version": "3.1.2",

      "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",

      "integrity": "sha512-/Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/call-bind": {

      "version": "1.0.7",

      "resolved": "https://registry.npmjs.org/call-bind/-/call-bind-1.0.7.tgz",

      "integrity": "sha512-GHTSNSYICQ7scH7sZ+M2rFopRoLh8t2bLSW6BbgrtLsahOIB5iyAVJf9GjWK3cYTDaMj4XdBpM1cA6pIS0Kv2w==",

      "license": "MIT",

      "dependencies": {

        "es-define-property": "^1.0.0",

        "es-errors": "^1.3.0",

        "function-bind": "^1.1.2",

        "get-intrinsic": "^1.2.4",

        "set-function-length": "^1.2.1"

      },

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/content-disposition": {

      "version": "0.5.4",

      "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",

      "integrity": "sha512-FveZTNuGw04cxlAiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevkw5Rqk+tSQ==",

      "dependencies": {

        "safe-buffer": "5.2.1"

      },

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/content-type": {

      "version": "1.0.5",

      "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.5.tgz",

      "integrity": "sha512-nTjqfcBFEipKdXCv4YDQWCfmcLZKm81ldF0pAopTvyrFGVbcR6P/VAAd5G7N+0tTr8QqiU0tFadD6FK4NtJwOA==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/cookie": {

      "version": "0.7.1",

      "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.1.tgz",

      "integrity": "sha512-6DnInpx7SJ2AK3+CTUE/ZM0vWTUboZCegxhC2xiIydHR9jNuTAASBrfEpHhiGOZw/nX51bHt6YQl8jsGo4y/0w==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/cookie-signature": {

      "version": "1.0.6",

      "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",

      "integrity": "sha512-QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQIrUgOWtHdNDtPiywmFbiS12VjotIXLrKM3orQ=="

    },

    "node\_modules/debug": {

      "version": "2.6.9",

      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",

      "integrity": "sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",

      "license": "MIT",

      "dependencies": {

        "ms": "2.0.0"

      }

    },

    "node\_modules/define-data-property": {

      "version": "1.1.4",

      "resolved": "https://registry.npmjs.org/define-data-property/-/define-data-property-1.1.4.tgz",

      "integrity": "sha512-rBMvIzlpA8v6E+SJZoo++HAYqsLrkg7MSfIinMPFhmkorw7X+dOXVJQs+QT69zGkzMyfDnIMN2Wid1+NbL3T+A==",

      "license": "MIT",

      "dependencies": {

        "es-define-property": "^1.0.0",

        "es-errors": "^1.3.0",

        "gopd": "^1.0.1"

      },

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/depd": {

      "version": "2.0.0",

      "resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",

      "integrity": "sha512-g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3fIh61/s/nPsfR6onyMwkCAR/OlC3yBC0lESvUoQEAssIrw==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/destroy": {

      "version": "1.2.0",

      "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",

      "integrity": "sha512-2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpweaVcPevC2l3KpjYCx3NypQQgaJg==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8",

        "npm": "1.2.8000 || >= 1.4.16"

      }

    },

    "node\_modules/dotenv": {

      "version": "16.3.1",

      "resolved": "https://registry.npmjs.org/dotenv/-/dotenv-16.3.1.tgz",

      "integrity": "sha512-IPzF4w4/Rd94bA9imS68tZBaYyBWSCE47V1RGuMrB94iyTOIEwRmVL2x/4An+6mETpLrKJ5hQkB8W4kFAadeIQ==",

      "engines": {

        "node": ">=12"

      },

      "funding": {

        "url": "https://github.com/motdotla/dotenv?sponsor=1"

      }

    },

    "node\_modules/ee-first": {

      "version": "1.1.1",

      "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",

      "integrity": "sha512-WMwm9LhRUo+WUaRN+vRuETqG89IgZphVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8lefD8KKfr5vY61brQlow==",

      "license": "MIT"

    },

    "node\_modules/encodeurl": {

      "version": "2.0.0",

      "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",

      "integrity": "sha512-Q0n9HRi4m6JuGIV1eFlmvJB7ZEVxu93IrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb31vhGgXnfmPNNXmxnOkRBrg==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/es-define-property": {

      "version": "1.0.0",

      "resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.0.tgz",

      "integrity": "sha512-jxayLKShrEqqzJ0eumQbVhTYQM27CfT1T35+gCgDFoL82JLsXqTJ76zv6A0YLOgEnLUMvLzsDsGIrl8NFpT2gQ==",

      "license": "MIT",

      "dependencies": {

        "get-intrinsic": "^1.2.4"

      },

      "engines": {

        "node": ">= 0.4"

      }

    },

    "node\_modules/es-errors": {

      "version": "1.3.0",

      "resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",

      "integrity": "sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzIlypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.4"

      }

    },

    "node\_modules/escape-html": {

      "version": "1.0.3",

      "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",

      "integrity": "sha512-NiSupZ4OeuGwr68lGIeym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxlJOMpijaKZVjAJrWrGs/6Jy8OMuyj9ow==",

      "license": "MIT"

    },

    "node\_modules/etag": {

      "version": "1.8.1",

      "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",

      "integrity": "sha512-aIL5Fx7mawVa300al2BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ2AS7kgVF/KEZWufBfdClMcPg==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/express": {

      "version": "4.21.1",

      "resolved": "https://registry.npmjs.org/express/-/express-4.21.1.tgz",

      "integrity": "sha512-YSFlK1Ee0/GC8QaO91tHcDxJiE/X4FbpAyQWkxAvG6AXCuR65YzK8ua6D9hvi/TzUfZMpc+BwuM1IPw8fmQBiQ==",

      "license": "MIT",

      "dependencies": {

        "accepts": "~1.3.8",

        "array-flatten": "1.1.1",

        "body-parser": "1.20.3",

        "content-disposition": "0.5.4",

        "content-type": "~1.0.4",

        "cookie": "0.7.1",

        "cookie-signature": "1.0.6",

        "debug": "2.6.9",

        "depd": "2.0.0",

        "encodeurl": "~2.0.0",

        "escape-html": "~1.0.3",

        "etag": "~1.8.1",

        "finalhandler": "1.3.1",

        "fresh": "0.5.2",

        "http-errors": "2.0.0",

        "merge-descriptors": "1.0.3",

        "methods": "~1.1.2",

        "on-finished": "2.4.1",

        "parseurl": "~1.3.3",

        "path-to-regexp": "0.1.10",

        "proxy-addr": "~2.0.7",

        "qs": "6.13.0",

        "range-parser": "~1.2.1",

        "safe-buffer": "5.2.1",

        "send": "0.19.0",

        "serve-static": "1.16.2",

        "setprototypeof": "1.2.0",

        "statuses": "2.0.1",

        "type-is": "~1.6.18",

        "utils-merge": "1.0.1",

        "vary": "~1.1.2"

      },

      "engines": {

        "node": ">= 0.10.0"

      }

    },

    "node\_modules/finalhandler": {

      "version": "1.3.1",

      "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.3.1.tgz",

      "integrity": "sha512-6BN9trH7bp3qvnrRyzsBz+g3lZxTNZTbVO2EV1CS0WIcDbawYVdYvGflME/9QP0h0pYlCDBCTjYa9nZzMDpyxQ==",

      "license": "MIT",

      "dependencies": {

        "debug": "2.6.9",

        "encodeurl": "~2.0.0",

        "escape-html": "~1.0.3",

        "on-finished": "2.4.1",

        "parseurl": "~1.3.3",

        "statuses": "2.0.1",

        "unpipe": "~1.0.0"

      },

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/forwarded": {

      "version": "0.2.0",

      "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",

      "integrity": "sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt6MT9aYow==",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/fresh": {

      "version": "0.5.2",

      "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",

      "integrity": "sha512-zJ2mQYM18rEFOudeV4GShTGIQ7RbzA7ozbU9I/XBpm7kqgMywgmylMwXHxZJmkVoYkna9d2pVXVXPdYTP9ej8Q==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/function-bind": {

      "version": "1.1.2",

      "resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",

      "integrity": "sha512-7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==",

      "license": "MIT",

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/get-intrinsic": {

      "version": "1.2.4",

      "resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.2.4.tgz",

      "integrity": "sha512-5uYhsJH8VJBTv7oslg4BznJYhDoRI6waYCxMmCdnTrcCrHA/fCFKoTFz2JKKE0HdDFUF7/oQuhzumXJK7paBRQ==",

      "license": "MIT",

      "dependencies": {

        "es-errors": "^1.3.0",

        "function-bind": "^1.1.2",

        "has-proto": "^1.0.1",

        "has-symbols": "^1.0.3",

        "hasown": "^2.0.0"

      },

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/gopd": {

      "version": "1.0.1",

      "resolved": "https://registry.npmjs.org/gopd/-/gopd-1.0.1.tgz",

      "integrity": "sha512-d65bNlIadxvpb/A2abVdlqKqV563juRnZ1Wtk6s1sIR8uNsXR70xqIzVqxVf1eTqDunwT2MkczEeaezCKTZhwA==",

      "license": "MIT",

      "dependencies": {

        "get-intrinsic": "^1.1.3"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/has-property-descriptors": {

      "version": "1.0.2",

      "resolved": "https://registry.npmjs.org/has-property-descriptors/-/has-property-descriptors-1.0.2.tgz",

      "integrity": "sha512-55JNKuIW+vq4Ke1BjOTjM2YctQIvCT7GFzHwmfZPGo5wnrgkid0YQtnAleFSqumZm4az3n2BS+erby5ipJdgrg==",

      "license": "MIT",

      "dependencies": {

        "es-define-property": "^1.0.0"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/has-proto": {

      "version": "1.0.3",

      "resolved": "https://registry.npmjs.org/has-proto/-/has-proto-1.0.3.tgz",

      "integrity": "sha512-SJ1amZAJUiZS+PhsVLf5tGydlaVB8EdFpaSO4gmiUKUOxk8qzn5AIy4ZeJUmh22znIdk/uMAUT2pl3FxzVUH+Q==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/has-symbols": {

      "version": "1.0.3",

      "resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.0.3.tgz",

      "integrity": "sha512-l3LCuF6MgDNwTDKkdYGEihYjt5pRPbEg46rtlmnSPlUbgmB8LOIrKJbYYFBSbnPaJexMKtiPO8hmeRjRz2Td+A==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/hasown": {

      "version": "2.0.2",

      "resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",

      "integrity": "sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+CyS8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7bfgSjkUMQ==",

      "license": "MIT",

      "dependencies": {

        "function-bind": "^1.1.2"

      },

      "engines": {

        "node": ">= 0.4"

      }

    },

    "node\_modules/http-errors": {

      "version": "2.0.0",

      "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",

      "integrity": "sha512-FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1plBdXgbyjv34pHTSb9zoeHMyDy33+DWy5Wt9Wo+TURtOYSQ==",

      "license": "MIT",

      "dependencies": {

        "depd": "2.0.0",

        "inherits": "2.0.4",

        "setprototypeof": "1.2.0",

        "statuses": "2.0.1",

        "toidentifier": "1.0.1"

      },

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/iconv-lite": {

      "version": "0.4.24",

      "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",

      "integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR31wwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj8/rA==",

      "license": "MIT",

      "dependencies": {

        "safer-buffer": ">= 2.1.2 < 3"

      },

      "engines": {

        "node": ">=0.10.0"

      }

    },

    "node\_modules/inherits": {

      "version": "2.0.4",

      "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",

      "integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4dYeZIQqewQ==",

      "license": "ISC"

    },

    "node\_modules/ipaddr.js": {

      "version": "1.9.1",

      "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",

      "integrity": "sha512-0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZlXSHBYXiYUPO3g==",

      "engines": {

        "node": ">= 0.10"

      }

    },

    "node\_modules/media-typer": {

      "version": "0.3.0",

      "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",

      "integrity": "sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOgb5O+a3KxfWapPQ==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/merge-descriptors": {

      "version": "1.0.3",

      "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.3.tgz",

      "integrity": "sha512-gaNvAS7TZ897/rVaZ0nMtAyxNyi/pdbjbAwUpFQpN70GqnVfOiXpeUUMKRBmzXaSQ8DdTX4/0ms62r2K+hE6mQ==",

      "license": "MIT",

      "funding": {

        "url": "https://github.com/sponsors/sindresorhus"

      }

    },

    "node\_modules/methods": {

      "version": "1.1.2",

      "resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",

      "integrity": "sha512-iclAHeNqNm68zFtnZ0e+1L2yUIdvzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/mime": {

      "version": "1.6.0",

      "resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",

      "integrity": "sha512-x0Vn8spI+wuJ1O6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOplvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==",

      "license": "MIT",

      "bin": {

        "mime": "cli.js"

      },

      "engines": {

        "node": ">=4"

      }

    },

    "node\_modules/mime-db": {

      "version": "1.52.0",

      "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",

      "integrity": "sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/mime-types": {

      "version": "2.1.35",

      "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",

      "integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",

      "dependencies": {

        "mime-db": "1.52.0"

      },

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/ms": {

      "version": "2.0.0",

      "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",

      "integrity": "sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",

      "license": "MIT"

    },

    "node\_modules/negotiator": {

      "version": "0.6.3",

      "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",

      "integrity": "sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGlQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdghQM4bBg==",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/object-inspect": {

      "version": "1.13.2",

      "resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.2.tgz",

      "integrity": "sha512-IRZSRuzJiynemAXPYtPe5BoI/RESNYR7TYm50MC5Mqbd3Jmw5y790sErYw3V6SryFJD64b74qQQs9wn5Bg/k3g==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/on-finished": {

      "version": "2.4.1",

      "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",

      "integrity": "sha512-oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/ClmuDr8Ch5+kg==",

      "license": "MIT",

      "dependencies": {

        "ee-first": "1.1.1"

      },

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/parseurl": {

      "version": "1.3.3",

      "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",

      "integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/path-to-regexp": {

      "version": "0.1.10",

      "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.10.tgz",

      "integrity": "sha512-7lf7qcQidTku0Gu3YDPc8DJ1q7OOucfa/BSsIwjuh56VU7katFvuM8hULfkwB3Fns/rsVF7PwPKVw1sl5KQS9w==",

      "license": "MIT"

    },

    "node\_modules/proxy-addr": {

      "version": "2.0.7",

      "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",

      "integrity": "sha512-llQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg==",

      "dependencies": {

        "forwarded": "0.2.0",

        "ipaddr.js": "1.9.1"

      },

      "engines": {

        "node": ">= 0.10"

      }

    },

    "node\_modules/qs": {

      "version": "6.13.0",

      "resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",

      "integrity": "sha512-+38qI9SOr8tfZ4QmJNplMUxqjbe7LKvvZgWdExBOmd+egZTtjLB67Gu0HRX3u/XOq7UU2Nx6nsjvS16Z9uwfpg==",

      "license": "BSD-3-Clause",

      "dependencies": {

        "side-channel": "^1.0.6"

      },

      "engines": {

        "node": ">=0.6"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/range-parser": {

      "version": "1.2.1",

      "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",

      "integrity": "sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAveqCBrwr/2yxQ5BVd/GTl5agOwSg==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/raw-body": {

      "version": "2.5.2",

      "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",

      "integrity": "sha512-8zGqypfENjCIqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loCVfA==",

      "license": "MIT",

      "dependencies": {

        "bytes": "3.1.2",

        "http-errors": "2.0.0",

        "iconv-lite": "0.4.24",

        "unpipe": "1.0.0"

      },

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/safe-buffer": {

      "version": "5.2.1",

      "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",

      "integrity": "sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ==",

      "funding": [

        {

          "type": "github",

          "url": "https://github.com/sponsors/feross"

        },

        {

          "type": "patreon",

          "url": "https://www.patreon.com/feross"

        },

        {

          "type": "consulting",

          "url": "https://feross.org/support"

        }

      ]

    },

    "node\_modules/safer-buffer": {

      "version": "2.1.2",

      "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",

      "integrity": "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==",

      "license": "MIT"

    },

    "node\_modules/send": {

      "version": "0.19.0",

      "resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",

      "integrity": "sha512-dW41u5VfLXu8SJh5bwRmyYUbAoSB3c9uQh6L8h/KtsFREPWpbX1lrljJo186Jc4nmci/sGUZ9a0a0J2zgfq2hw==",

      "license": "MIT",

      "dependencies": {

        "debug": "2.6.9",

        "depd": "2.0.0",

        "destroy": "1.2.0",

        "encodeurl": "~1.0.2",

        "escape-html": "~1.0.3",

        "etag": "~1.8.1",

        "fresh": "0.5.2",

        "http-errors": "2.0.0",

        "mime": "1.6.0",

        "ms": "2.1.3",

        "on-finished": "2.4.1",

        "range-parser": "~1.2.1",

        "statuses": "2.0.1"

      },

      "engines": {

        "node": ">= 0.8.0"

      }

    },

    "node\_modules/send/node\_modules/encodeurl": {

      "version": "1.0.2",

      "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",

      "integrity": "sha512-TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAmJEdAOx0w==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/send/node\_modules/ms": {

      "version": "2.1.3",

      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",

      "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==",

      "license": "MIT"

    },

    "node\_modules/serve-static": {

      "version": "1.16.2",

      "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",

      "integrity": "sha512-VqpjJZKadQB/PEbEwvFdO43Ax5dFBZ2UECszz8bQ7pi7wt//PWe1P6MN7eCnjsatYtBT6EuiClbjSWP2WrIoTw==",

      "license": "MIT",

      "dependencies": {

        "encodeurl": "~2.0.0",

        "escape-html": "~1.0.3",

        "parseurl": "~1.3.3",

        "send": "0.19.0"

      },

      "engines": {

        "node": ">= 0.8.0"

      }

    },

    "node\_modules/set-function-length": {

      "version": "1.2.2",

      "resolved": "https://registry.npmjs.org/set-function-length/-/set-function-length-1.2.2.tgz",

      "integrity": "sha512-pgRc4hJ4/sNjWCSS9AmnS40x3bNMDTknHgL5UaMBTMyJnU90EgWh1Rz+MC9eFu4BuN/UwZjKQuY/1v3rM7HMfg==",

      "license": "MIT",

      "dependencies": {

        "define-data-property": "^1.1.4",

        "es-errors": "^1.3.0",

        "function-bind": "^1.1.2",

        "get-intrinsic": "^1.2.4",

        "gopd": "^1.0.1",

        "has-property-descriptors": "^1.0.2"

      },

      "engines": {

        "node": ">= 0.4"

      }

    },

    "node\_modules/setprototypeof": {

      "version": "1.2.0",

      "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",

      "integrity": "sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNwYMJpw==",

      "license": "ISC"

    },

    "node\_modules/side-channel": {

      "version": "1.0.6",

      "resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.0.6.tgz",

      "integrity": "sha512-fDW/EZ6Q9RiO8eFG8Hj+7u/oW+XrPTIChwCOM2+th2A6OblDtYYIpve9m+KvI9Z4C9qSEXlaGR6bTEYHReuglA==",

      "license": "MIT",

      "dependencies": {

        "call-bind": "^1.0.7",

        "es-errors": "^1.3.0",

        "get-intrinsic": "^1.2.4",

        "object-inspect": "^1.13.1"

      },

      "engines": {

        "node": ">= 0.4"

      },

      "funding": {

        "url": "https://github.com/sponsors/ljharb"

      }

    },

    "node\_modules/statuses": {

      "version": "2.0.1",

      "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",

      "integrity": "sha512-RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mTCbkdQ==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/toidentifier": {

      "version": "1.0.1",

      "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",

      "integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA==",

      "license": "MIT",

      "engines": {

        "node": ">=0.6"

      }

    },

    "node\_modules/type-is": {

      "version": "1.6.18",

      "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",

      "integrity": "sha512-TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQBy8g==",

      "license": "MIT",

      "dependencies": {

        "media-typer": "0.3.0",

        "mime-types": "~2.1.24"

      },

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/unpipe": {

      "version": "1.0.0",

      "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",

      "integrity": "sha512-pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztKQ==",

      "license": "MIT",

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/utils-merge": {

      "version": "1.0.1",

      "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",

      "integrity": "sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsHHoED/ztMtkv/VoYTYyShUn81hA==",

      "engines": {

        "node": ">= 0.4.0"

      }

    },

    "node\_modules/vary": {

      "version": "1.1.2",

      "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",

      "integrity": "sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr3eIZ/PXqg==",

      "engines": {

        "node": ">= 0.8"

      }

    }

  }

}

**package.json:**

{

  "name": "gemini\_chatbot",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "@google/generative-ai": "^0.1.3",

    "dotenv": "^16.3.1",

    "express": "^4.21.1"

  }

}

**README.md:**

# coding-money-chatbot

## Installation

1. npm install

2. create .env file and add your API key as:

     API\_KEY="Paste API Key here"

**server.js:**

const express = require('express');

const { GoogleGenerativeAI, HarmCategory, HarmBlockThreshold } = require('@google/generative-ai');

const dotenv = require('dotenv').config();

const app = express();

const port = process.env.PORT || 3000;

app.use(express.json());

const MODEL\_NAME = "gemini-pro";

const API\_KEY = process.env.API\_KEY;

// In-memory object to store conversation history for each user session

let conversationHistory = [];

async function runChat(userInput) {

  const genAI = new GoogleGenerativeAI(API\_KEY);

  const model = genAI.getGenerativeModel({ model: MODEL\_NAME });

  const generationConfig = {

    temperature: 0.9,

    topK: 1,

    topP: 1,

    maxOutputTokens: 1000,

  };

  const safetySettings = [

    {

      category: HarmCategory.HARM\_CATEGORY\_HARASSMENT,

      threshold: HarmBlockThreshold.BLOCK\_MEDIUM\_AND\_ABOVE,

    },

    // ... other safety settings

  ];

  // Add the user's input to the conversation history

  conversationHistory.push({

    role: "user",

    parts: [{ text: userInput }],

  });

  const chat = model.startChat({

    generationConfig,

    safetySettings,

    history: conversationHistory,

  });

  const result = await chat.sendMessage(userInput);

  const response = result.response;

  // Add the model's response to the conversation history

  conversationHistory.push({

    role: "model",

    parts: [{ text: response.text() }],

  });

  // return response.text();

  return response.candidates[0]?.content?.parts[0]?.text || "No response";

}

app.get('/', (req, res) => {

  res.sendFile(\_\_dirname + '/index.html');

});

app.get('/loader.gif', (req, res) => {

  res.sendFile(\_\_dirname + '/loader.gif');

});

app.post('/chat', async (req, res) => {

  try {

    const userInput = req.body?.userInput;

    console.log('Incoming /chat request:', userInput);

    if (!userInput) {

      return res.status(400).json({ error: 'Invalid request body' });

    }

    const response = await runChat(userInput);

    res.json({ response });

  } catch (error) {

    console.error('Error in chat endpoint:', error);

    res.status(500).json({ error: 'Internal Server Error' });

  }

});

app.listen(port, () => {

  console.log(`Server listening on port ${port}`);

});

**Model:**

**code.ipynb:**

import kagglehub

# Download the latest version of the dataset

path = kagglehub.dataset\_download("salmansajid05/oral-diseases")

# Print the path to the dataset files

print("Path to dataset files:", path)

# Set the root directory for the dataset

root = path

import os

import torch

import shutil

import numpy as np

from glob import glob

from PIL import Image

from torch.utils.data import random\_split, Dataset, DataLoader

from torchvision import transforms as T

torch.manual\_seed(2024)

class CustomDataset(Dataset):

    def \_\_init\_\_(self, root, transformations=None):

        self.transformations = transformations

        self.im\_paths = sorted(glob(f"{root}/\*/\*/\*.jpg"))

        self.cls\_names, self.cls\_counts, count = {}, {}, 0

        for idx, im\_path in enumerate(self.im\_paths):

            class\_name = self.get\_class(im\_path)

            if class\_name not in self.cls\_names:

                self.cls\_names[class\_name] = count

                self.cls\_counts[class\_name] = 1

                count += 1

            else:

                self.cls\_counts[class\_name] += 1

    def get\_class(self, path):

        # Extract only the class name without the full path

        return os.path.basename(os.path.dirname(path))

    def \_\_len\_\_(self):

        return len(self.im\_paths)

    def \_\_getitem\_\_(self, idx):

        im\_path = self.im\_paths[idx]

        im = Image.open(im\_path).convert("RGB")

        gt = self.cls\_names[self.get\_class(im\_path)]

        if self.transformations is not None:

            im = self.transformations(im)

        return im, gt

def get\_dls(root, transformations, bs, split=[0.9, 0.05, 0.05], ns=4):

    ds = CustomDataset(root=root, transformations=transformations)

    total\_len = len(ds)

    tr\_len = int(total\_len \* split[0])

    vl\_len = int(total\_len \* split[1])

    ts\_len = total\_len - (tr\_len + vl\_len)

    tr\_ds, vl\_ds, ts\_ds = random\_split(dataset=ds, lengths=[tr\_len, vl\_len, ts\_len])

    tr\_dl = DataLoader(tr\_ds, batch\_size=bs, shuffle=True, num\_workers=ns)

    val\_dl = DataLoader(vl\_ds, batch\_size=bs, shuffle=False, num\_workers=ns)

    ts\_dl = DataLoader(ts\_ds, batch\_size=1, shuffle=False, num\_workers=ns)

    return tr\_dl, val\_dl, ts\_dl, ds.cls\_names

# Define the root path and transformations

root = r"C:\Users\shafe\.cache\kagglehub\datasets\salmansajid05\oral-diseases\versions\3"

mean, std, im\_size = [0.485, 0.456, 0.406], [0.229, 0.224, 0.225], 224

tfs = T.Compose([

    T.Resize((im\_size, im\_size)),

    T.ToTensor(),

    T.Normalize(mean=mean, std=std)

])

# Get the data loaders

tr\_dl, val\_dl, ts\_dl, classes = get\_dls(root=root, transformations=tfs, bs=32)

# Print some information about the data loaders

print(f"Number of batches in training DataLoader: {len(tr\_dl)}")

print(f"Number of batches in validation DataLoader: {len(val\_dl)}")

print(f"Number of batches in test DataLoader: {len(ts\_dl)}")

print(f"Classes: {classes}")

import random

from matplotlib import pyplot as plt

# Ensure these imports are available from Step 1

# If you've run Step 1, these should already be imported

from torchvision import transforms as T

# Function to convert tensor back to image for visualization

def tensor\_2\_im(t, t\_type="rgb"):

    gray\_tfs = T.Compose([

        T.Normalize(mean=[0.], std=[1/0.5]),

        T.Normalize(mean=[-0.5], std=[1])

    ])

    rgb\_tfs = T.Compose([

        T.Normalize(mean=[0., 0., 0.], std=[1/0.229, 1/0.224, 1/0.225]),

        T.Normalize(mean=[-0.485, -0.456, -0.406], std=[1., 1., 1.])

    ])

    invTrans = gray\_tfs if t\_type == "gray" else rgb\_tfs

    return (invTrans(t) \* 255).detach().squeeze().cpu().permute(1, 2, 0).numpy().astype(np.uint8) if t\_type == "gray" else (invTrans(t) \* 255).detach().cpu().permute(1, 2, 0).numpy().astype(np.uint8)

# Visualization function

def visualize(data, n\_ims, rows, cmap=None, cls\_names=None):

    assert cmap in ["rgb", "gray"], "Rasmni oq-qora yoki rangli ekanini aniqlashtirib bering!"

    if cmap == "rgb":

        cmap = "viridis"

    plt.figure(figsize=(20, 10))

    indekslar = [random.randint(0, len(data) - 1) for \_ in range(n\_ims)]

    for idx, indeks in enumerate(indekslar):

        im, gt = data[indeks]

        plt.subplot(rows, n\_ims // rows, idx + 1)

        if cmap:

            plt.imshow(tensor\_2\_im(im, cmap), cmap=cmap)

        else:

            plt.imshow(tensor\_2\_im(im))

        plt.axis('off')

        if cls\_names is not None:

            plt.title(f"GT -> {cls\_names[int(gt)]}")

        else:

            plt.title(f"GT -> {gt}")

    plt.show()

# Visualize training, validation, and test datasets

# These variables (tr\_dl, val\_dl, ts\_dl, classes) should be available from Step 1

visualize(tr\_dl.dataset, 20, 4, "rgb", list(classes.keys()))

visualize(val\_dl.dataset, 20, 4, "rgb", list(classes.keys()))

visualize(ts\_dl.dataset, 20, 4, "rgb", list(classes.keys()))

import numpy as np

from matplotlib import pyplot as plt

def data\_analysis(root, transformations):

    ds = CustomDataset(root=root, transformations=transformations)

    cls\_counts = ds.cls\_counts

    cls\_names = list(cls\_counts.keys())

    counts = list(cls\_counts.values())

    # Expected counts from the dataset description

    expected\_counts = {

        "Calculus": 1296,

        "Data Caries": 2382,

        "Gingivitis": 2349,

        "Tooth Discoloration": 1834,

        "Mouth Ulcer": 2541

    }

    # Verify class counts

    print("Verifying class counts against expected values:")

    for cls\_name, count in cls\_counts.items():

        expected = expected\_counts.get(cls\_name, "Not specified")

        print(f"{cls\_name}: {count} images (Expected: {expected})")

        if cls\_name in expected\_counts and count != expected\_counts[cls\_name]:

            print(f"Warning: Count mismatch for {cls\_name}!")

    # Create a donut chart

    plt.figure(figsize=(10, 10))

    plt.pie(counts, labels=cls\_names, autopct='%1.1f%%', startangle=140, colors=plt.cm.Paired(np.arange(len(counts))), wedgeprops=dict(width=0.4))

    plt.title("Class Distribution in Oral Diseases Dataset (Donut Chart)")

    plt.show()

# Perform data analysis

data\_analysis(root=root, transformations=tfs)

import kagglehub

import os

# Download the latest version of the dataset

dataset\_name = "imtkaggleteam/dental-radiography"

print(f"Downloading dataset {dataset\_name}...")

path = kagglehub.dataset\_download(dataset\_name)

print("Path to dataset files:", path)

# Ensure the dataset is in the expected directory

expected\_path = r"C:\Users\shafe\.cache\kagglehub\datasets\imtkaggleteam\dental-radiography"

if os.path.realpath(path) != os.path.realpath(expected\_path):

    print(f"Warning: Dataset downloaded to {path}, but expected at {expected\_path}.")

    print("You may need to update the training script's root path accordingly.")

import os

# Path where the dataset was downloaded

dataset\_path = r"C:\Users\shafe\.cache\kagglehub\datasets\imtkaggleteam\dental-radiography\versions\1"

# Recursively walk through the directory and print its structure

print("Exploring dataset structure:")

for root, dirs, files in os.walk(dataset\_path):

    level = root.replace(dataset\_path, '').count(os.sep)

    indent = ' ' \* 4 \* level

    print(f"{indent}{os.path.basename(root)}/")

    for f in files:

        print(f"{indent}    {f}")

cd C:\Users\shafe\SmartDentAi-project

conda activate base

jupyter notebook

import os

import shutil

import random

import pandas as pd

# Path where the dataset was downloaded

dataset\_path = r"C:\Users\shafe\.cache\kagglehub\datasets\imtkaggleteam\dental-radiography\versions\1"

# Create a temporary directory to combine all images

temp\_dir = os.path.join(dataset\_path, "temp\_all\_images")

os.makedirs(temp\_dir, exist\_ok=True)

# Combine annotations from all splits

all\_annotations = []

splits = ["train", "val", "test"]

for split in splits:

    split\_path = os.path.join(dataset\_path, split)

    if os.path.exists(split\_path):

        annotations\_path = os.path.join(split\_path, "\_annotations.csv")

        if os.path.exists(annotations\_path):

            df = pd.read\_csv(annotations\_path)

            all\_annotations.append(df)

        # Move images to temp directory

        for img in os.listdir(split\_path):

            if img.endswith(".jpg"):

                src = os.path.join(split\_path, img)

                dst = os.path.join(temp\_dir, img)

                shutil.move(src, dst)

# Combine all annotations into a single DataFrame

if all\_annotations:

    all\_annotations\_df = pd.concat(all\_annotations, ignore\_index=True)

else:

    raise FileNotFoundError("No \_annotations.csv files found in any split!")

# Get list of all images

all\_images = [f for f in os.listdir(temp\_dir) if f.endswith(".jpg")]

print(f"Total images combined: {len(all\_images)}")

# Define split sizes explicitly for 70/20/10

total\_images = len(all\_images)

train\_size = 804  # 70% of 1148

val\_size = 230    # 20% of 1148

test\_size = 114   # 10% of 1148

if total\_images != (train\_size + val\_size + test\_size):

    raise ValueError(f"Split sizes do not add up to total images: {train\_size} + {val\_size} + {test\_size} != {total\_images}")

print(f"Train size: {train\_size}")

print(f"Val size: {val\_size}")

print(f"Test size: {test\_size}")

# Randomly split the images

random.seed(2024)  # For reproducibility

random.shuffle(all\_images)

train\_images = all\_images[:train\_size]

val\_images = all\_images[train\_size:train\_size + val\_size]

test\_images = all\_images[train\_size + val\_size:]

# Create directories for each split

for split in ["train", "val", "test"]:

    split\_path = os.path.join(dataset\_path, split)

    # Clear existing directory

    if os.path.exists(split\_path):

        shutil.rmtree(split\_path)

    os.makedirs(split\_path, exist\_ok=True)

# Move images to their respective splits

def move\_images(image\_list, split):

    split\_path = os.path.join(dataset\_path, split)

    for img in image\_list:

        src = os.path.join(temp\_dir, img)

        dst = os.path.join(split\_path, img)

        shutil.move(src, dst)

move\_images(train\_images, "train")

move\_images(val\_images, "val")

move\_images(test\_images, "test")

# Split annotations based on the images in each split

train\_annotations = all\_annotations\_df[all\_annotations\_df["filename"].isin(train\_images)]

val\_annotations = all\_annotations\_df[all\_annotations\_df["filename"].isin(val\_images)]

test\_annotations = all\_annotations\_df[all\_annotations\_df["filename"].isin(test\_images)]

# Save the updated annotations

train\_annotations.to\_csv(os.path.join(dataset\_path, "train", "\_annotations.csv"), index=False)

val\_annotations.to\_csv(os.path.join(dataset\_path, "val", "\_annotations.csv"), index=False)

test\_annotations.to\_csv(os.path.join(dataset\_path, "test", "\_annotations.csv"), index=False)

print(f"Saved {len(train\_annotations)} annotations to train/\_annotations.csv")

print(f"Saved {len(val\_annotations)} annotations to val/\_annotations.csv")

print(f"Saved {len(test\_annotations)} annotations to test/\_annotations.csv")

# Clean up the temporary directory

shutil.rmtree(temp\_dir)

# Verify the updated splits

print("\nUpdated dataset structure:")

for split in ["train", "val", "test"]:

    split\_path = os.path.join(dataset\_path, split)

    if os.path.exists(split\_path):

        images = [f for f in os.listdir(split\_path) if f.endswith(".jpg")]

        print(f"Contents of {split} directory:")

        print(f"Number of images: {len(images)}")

        print("Sample images:", images[:5] if len(images) > 5 else images)

        # Check for \_annotations.csv

        annotations\_file = os.path.join(split\_path, "\_annotations.csv")

        if os.path.exists(annotations\_file):

            annotations\_df = pd.read\_csv(annotations\_file)

            print(f"Number of annotations: {len(annotations\_df)}")

            print("Sample annotations:")

            print(annotations\_df.head())

        else:

            print(f"\_annotations.csv not found in {split} directory!")

    else:

        print(f"{split} directory not found!")

**plot\_metrics.py:**

# plot\_metrics.py

import pandas as pd

import matplotlib.pyplot as plt

# Path to the saved metrics

metrics\_path = r"C:\Users\shafe\SmartDentAi-project\training\_metrics.csv"

# Load the metrics

metrics\_df = pd.read\_csv(metrics\_path)

# Plot the metrics

plt.figure(figsize=(15, 10))

# Plot Loss

plt.subplot(3, 1, 1)

plt.plot(metrics\_df["Epoch"], metrics\_df["Train Loss"], label="Train Loss", marker="o")

plt.plot(metrics\_df["Epoch"], metrics\_df["Val Loss"], label="Val Loss", marker="o")

plt.title("Training and Validation Loss")

plt.xlabel("Epoch")

plt.ylabel("Loss")

plt.legend()

plt.grid()

# Plot Accuracy

plt.subplot(3, 1, 2)

plt.plot(metrics\_df["Epoch"], metrics\_df["Train Accuracy"], label="Train Accuracy", marker="o")

plt.plot(metrics\_df["Epoch"], metrics\_df["Val Accuracy"], label="Val Accuracy", marker="o")

plt.title("Training and Validation Accuracy")

plt.xlabel("Epoch")

plt.ylabel("Accuracy")

plt.legend()

plt.grid()

# Plot F1-Score

plt.subplot(3, 1, 3)

plt.plot(metrics\_df["Epoch"], metrics\_df["Train F1"], label="Train F1", marker="o")

plt.plot(metrics\_df["Epoch"], metrics\_df["Val F1"], label="Val F1", marker="o")

plt.title("Training and Validation F1-Score")

plt.xlabel("Epoch")

plt.ylabel("F1-Score")

plt.legend()

plt.grid()

plt.tight\_layout()

plt.savefig(r"C:\Users\shafe\SmartDentAi-project\learning\_curves\_90percent.png")

plt.show()

**train\_dental\_radiography.py:**

# train\_dental\_radiography.py

# Suppress the huggingface\_hub symlink warning

import os

os.environ["HF\_HUB\_DISABLE\_SYMLINKS\_WARNING"] = "1"

import torch

import numpy as np

from PIL import Image

from torch.utils.data import Dataset, DataLoader

from torchvision import transforms as T

import timm

import torchmetrics

from tqdm import tqdm

import kagglehub

import pandas as pd

import cv2

# Set random seed for reproducibility

torch.manual\_seed(2024)

# Custom CLAHE transform

class CLAHETransform:

    def \_\_init\_\_(self, clip\_limit=2.0, tile\_grid\_size=(8, 8)):

        self.clip\_limit = clip\_limit

        self.tile\_grid\_size = tile\_grid\_size

    def \_\_call\_\_(self, img):

        # Convert PIL Image to numpy array

        img\_np = np.array(img)

        # Convert to grayscale if RGB

        if len(img\_np.shape) == 3:

            img\_np = cv2.cvtColor(img\_np, cv2.COLOR\_RGB2GRAY)

        # Apply CLAHE

        clahe = cv2.createCLAHE(clipLimit=self.clip\_limit, tileGridSize=self.tile\_grid\_size)

        img\_clahe = clahe.apply(img\_np)

        # Convert back to RGB by duplicating channels

        img\_clahe = cv2.cvtColor(img\_clahe, cv2.COLOR\_GRAY2RGB)

        # Convert back to PIL Image

        return Image.fromarray(img\_clahe)

# Custom Dataset for Dental Radiography (Multi-Label Classification)

class CustomDataset(Dataset):

    def \_\_init\_\_(self, root, split, transformations=None, oversample=False):

        self.transformations = transformations

        self.root = root

        self.split = split.lower()

        # Adjust split name (val might be validation)

        split\_dir = self.split

        if self.split == "val":

            split\_path\_val = os.path.join(root, "val")

            split\_path\_validation = os.path.join(root, "validation")

            if os.path.exists(split\_path\_validation):

                split\_dir = "validation"

            elif not os.path.exists(split\_path\_val):

                raise FileNotFoundError("Neither 'val' nor 'validation' directory found")

        # Load annotations

        annotations\_path = os.path.join(root, split\_dir, "\_annotations.csv")

        if not os.path.exists(annotations\_path):

            raise FileNotFoundError(f"\_annotations.csv not found at {annotations\_path}")

        df = pd.read\_csv(annotations\_path)

        # Group annotations by filename to handle multiple labels per image

        self.image\_data = df.groupby("filename")["class"].apply(list).reset\_index()

        # Define classes

        self.cls\_names = sorted(df["class"].unique())  # ['Cavity', 'Fillings', 'Impacted Tooth', 'Implant']

        self.cls\_idx = {cls: idx for idx, cls in enumerate(self.cls\_names)}

        # Load image paths and labels

        self.im\_paths = []

        self.labels = []

        split\_path = os.path.join(root, split\_dir)

        for \_, row in self.image\_data.iterrows():

            image\_path = os.path.join(split\_path, row["filename"])

            if os.path.exists(image\_path):

                self.im\_paths.append(image\_path)

                # Create multi-label vector (e.g., [0, 1, 0, 1] for 4 classes)

                label\_vector = torch.zeros(len(self.cls\_names))

                for cls in row["class"]:

                    label\_vector[self.cls\_idx[cls]] = 1

                self.labels.append(label\_vector)

        print(f"Loaded {len(self.im\_paths)} images for {split} split before oversampling")

        # Oversample minority classes (only for training split)

        if oversample and self.split == "train":

            # Compute class frequencies

            cls\_counts = {cls: 0 for cls in self.cls\_names}

            for labels in self.labels:

                for idx, val in enumerate(labels):

                    if val == 1:

                        cls\_counts[self.cls\_names[idx]] += 1

            # Find the maximum class frequency

            max\_count = max(cls\_counts.values())

            oversampled\_paths = []

            oversampled\_labels = []

            for im\_path, label in zip(self.im\_paths, self.labels):

                oversampled\_paths.append(im\_path)

                oversampled\_labels.append(label)

                # Oversample based on the rarest class present in this image

                min\_class\_count = float('inf')

                for idx, val in enumerate(label):

                    if val == 1 and cls\_counts[self.cls\_names[idx]] < min\_class\_count:

                        min\_class\_count = cls\_counts[self.cls\_names[idx]]

                if min\_class\_count < max\_count:

                    # Calculate how many times to replicate this sample

                    num\_replicates = int(max\_count / min\_class\_count) - 1

                    for \_ in range(num\_replicates):

                        oversampled\_paths.append(im\_path)

                        oversampled\_labels.append(label)

            self.im\_paths = oversampled\_paths

            self.labels = oversampled\_labels

            print(f"After oversampling, {len(self.im\_paths)} images for {split} split")

        # Class counts for computing class weights

        self.cls\_counts = {cls: 0 for cls in self.cls\_names}

        for labels in self.labels:

            for idx, val in enumerate(labels):

                if val == 1:

                    self.cls\_counts[self.cls\_names[idx]] += 1

    def \_\_len\_\_(self):

        return len(self.im\_paths)

    def \_\_getitem\_\_(self, idx):

        im\_path = self.im\_paths[idx]

        try:

            im = Image.open(im\_path).convert("RGB")

        except Exception as e:

            print(f"Error loading image {im\_path}: {e}")

            return None, None

        label = self.labels[idx]

        if self.transformations is not None:

            im = self.transformations(im)

        return im, label

    def get\_class\_weights(self):

        # Compute class weights for imbalanced classes

        total\_samples = len(self.labels)

        num\_classes = len(self.cls\_names)

        weights = []

        for cls in self.cls\_names:

            pos = self.cls\_counts[cls]

            neg = total\_samples - pos

            if pos > 0:

                weight = (neg / pos) \* 2  # More aggressive weighting

            else:

                weight = 1.0

            weights.append(weight)

        return torch.tensor(weights)

def get\_dls(root, transformations, bs, ns=0):

    # Load datasets for train, validation, and test splits

    train\_ds = CustomDataset(root=root, split="train", transformations=transformations, oversample=True)

    val\_ds = CustomDataset(root=root, split="val", transformations=transformations, oversample=False)

    test\_ds = CustomDataset(root=root, split="test", transformations=transformations, oversample=False)

    # Store cls\_names before filtering

    cls\_names = train\_ds.cls\_names

    # Filter out None entries (invalid images)

    train\_data = [item for item in train\_ds if item[0] is not None]

    val\_data = [item for item in val\_ds if item[0] is not None]

    test\_data = [item for item in test\_ds if item[0] is not None]

    # Create DataLoaders

    tr\_dl = DataLoader(train\_data, batch\_size=bs, shuffle=True, num\_workers=ns)

    val\_dl = DataLoader(val\_data, batch\_size=bs, shuffle=False, num\_workers=ns)

    ts\_dl = DataLoader(test\_data, batch\_size=1, shuffle=False, num\_workers=ns)

    # Get class weights from the training dataset

    class\_weights = train\_ds.get\_class\_weights()

    return tr\_dl, val\_dl, ts\_dl, cls\_names, class\_weights

# Training Functions

def train\_setup(m, num\_classes, class\_weights):

    device = "cuda" if torch.cuda.is\_available() else "cpu"

    print(f"Using device: {device}")

    # Define the optimizer

    optimizer = torch.optim.Adam(params=m.parameters(), lr=1e-4, weight\_decay=1e-4)  # Increased lr, reduced weight decay

    # Define the scheduler

    scheduler = torch.optim.lr\_scheduler.ReduceLROnPlateau(optimizer, mode='min', factor=0.1, patience=5, verbose=True)

    # Use BCEWithLogitsLoss with label smoothing

    class SmoothBCEWithLogitsLoss(torch.nn.Module):

        def \_\_init\_\_(self, pos\_weight=None, smoothing=0.05):

            super().\_\_init\_\_()

            self.pos\_weight = pos\_weight

            self.smoothing = smoothing

        def forward(self, pred, target):

            target = target.float()

            target = target \* (1 - 2 \* self.smoothing) + self.smoothing  # Label smoothing

            return torch.nn.functional.binary\_cross\_entropy\_with\_logits(pred, target, pos\_weight=self.pos\_weight)

    loss\_fn = SmoothBCEWithLogitsLoss(pos\_weight=class\_weights.to(device), smoothing=0.05)

    return (

        m.to(device).eval(),

        100,

        device,

        loss\_fn,

        optimizer,

        scheduler

    )

def to\_device(batch, device):

    return batch[0].to(device), batch[1].to(device)

def get\_metrics(model, ims, gts, loss\_fn, epoch\_loss, epoch\_acc, epoch\_f1, f1\_score):

    preds = model(ims)

    preds = torch.sigmoid(preds)  # Sigmoid for multi-label classification

    loss = loss\_fn(preds, gts)

    if torch.isnan(loss) or torch.isinf(loss):

        print("Invalid loss detected!")

        return None, epoch\_loss, epoch\_acc, epoch\_f1, 0

    # Per-label accuracy: proportion of correct predictions per label

    pred\_labels = (preds > 0.5).float()

    correct = (pred\_labels == gts).float().sum().item()

    total = gts.numel()  # Total number of predictions (batch\_size \* num\_classes)

    return (

        loss,

        epoch\_loss + loss.item(),

        epoch\_acc + correct,

        epoch\_f1 + f1\_score(pred\_labels, gts),

        total

    )

def main():

    try:

        # Clear GPU memory if using GPU

        if torch.cuda.is\_available():

            torch.cuda.empty\_cache()

        # Download the dataset using kagglehub

        dataset\_name = "imtkaggleteam/dental-radiography"

        print(f"Ensuring dataset {dataset\_name} is downloaded...")

        root = kagglehub.dataset\_download(dataset\_name)

        print(f"Dataset path: {root}")

        # Define transformations with enhanced data augmentation

        mean, std, im\_size = [0.485, 0.456, 0.406], [0.229, 0.224, 0.225], 224

        train\_tfs = T.Compose([

            CLAHETransform(),  # Apply CLAHE

            T.RandomHorizontalFlip(p=0.5),

            T.RandomVerticalFlip(p=0.5),

            T.RandomRotation(degrees=45),

            T.RandomAffine(degrees=0, scale=(0.7, 1.3), shear=10),

            T.ColorJitter(brightness=0.3, contrast=0.3, saturation=0.3),

            T.RandomApply([T.GaussianBlur(kernel\_size=3)], p=0.2),

            T.Resize((im\_size, im\_size)),

            T.ToTensor(),

            T.Normalize(mean=mean, std=std)

        ])

        val\_test\_tfs = T.Compose([

            CLAHETransform(),

            T.Resize((im\_size, im\_size)),

            T.ToTensor(),

            T.Normalize(mean=mean, std=std)

        ])

        # Get the data loaders and class weights

        tr\_dl, val\_dl, ts\_dl, classes, class\_weights = get\_dls(root=root, transformations=train\_tfs, bs=16, ns=0)

        \_, val\_dl, ts\_dl, \_, \_ = get\_dls(root=root, transformations=val\_test\_tfs, bs=16, ns=0)

        # Print some information about the data loaders

        print(f"Number of batches in training DataLoader: {len(tr\_dl)}")

        print(f"Number of batches in validation DataLoader: {len(val\_dl)}")

        print(f"Number of batches in test DataLoader: {len(ts\_dl)}")

        print(f"Classes: {classes}")

        print(f"Class weights: {class\_weights}")

        # Initialize the DenseNet201 model with adjusted dropout

        m = timm.create\_model("densenet201", pretrained=True, num\_classes=len(classes), drop\_rate=0.4)

        # Training setup

        m, epochs, device, loss\_fn, optimizer, scheduler = train\_setup(m, len(classes), class\_weights)

        f1\_score = torchmetrics.F1Score(task="multilabel", num\_labels=len(classes)).to(device)

        save\_prefix, save\_dir = "dental\_radiography", r"C:\Users\shafe\SmartDentAi-project"

        print("Start training...")

        # Training loop variables

        best\_acc, threshold, not\_improved, patience = 0.0, 0.01, 0, 20  # Increased patience

        tr\_losses, val\_losses, tr\_accs, val\_accs, tr\_f1s, val\_f1s = [], [], [], [], [], []

        for epoch in range(epochs):

            print(f"Starting epoch {epoch + 1}/{epochs}...")

            # Training phase

            epoch\_loss, epoch\_acc, epoch\_f1, total\_preds = 0, 0, 0, 0

            m.train()

            for idx, batch in tqdm(enumerate(tr\_dl), total=len(tr\_dl)):

                ims, gts = to\_device(batch, device)

                metrics = get\_metrics(m, ims, gts, loss\_fn, epoch\_loss, epoch\_acc, epoch\_f1, f1\_score)

                if metrics[0] is None:

                    print(f"Stopping epoch {epoch + 1} due to invalid loss.")

                    break

                loss, epoch\_loss, epoch\_acc, epoch\_f1, total = metrics

                optimizer.zero\_grad()

                loss.backward()

                torch.nn.utils.clip\_grad\_norm\_(m.parameters(), max\_norm=1.0)

                optimizer.step()

                total\_preds += total

            tr\_loss\_to\_track = epoch\_loss / len(tr\_dl)

            tr\_acc\_to\_track = epoch\_acc / total\_preds

            tr\_f1\_to\_track = epoch\_f1 / len(tr\_dl)

            tr\_losses.append(tr\_loss\_to\_track)

            tr\_accs.append(tr\_acc\_to\_track)

            tr\_f1s.append(tr\_f1\_to\_track.cpu().item() if torch.is\_tensor(tr\_f1\_to\_track) else tr\_f1\_to\_track)

            print(f"{epoch + 1}-epoch train process is completed!")

            print(f"{epoch + 1}-epoch train loss          -> {tr\_loss\_to\_track:.3f}")

            print(f"{epoch + 1}-epoch train accuracy      -> {tr\_acc\_to\_track:.3f}")

            print(f"{epoch + 1}-epoch train f1-score      -> {tr\_f1\_to\_track:.3f}")

            # Validation phase

            print(f"Starting validation for epoch {epoch + 1}...")

            m.eval()

            with torch.no\_grad():

                val\_epoch\_loss, val\_epoch\_acc, val\_epoch\_f1, val\_total\_preds = 0, 0, 0, 0

                for idx, batch in enumerate(val\_dl):

                    ims, gts = to\_device(batch, device)

                    metrics = get\_metrics(m, ims, gts, loss\_fn, val\_epoch\_loss, val\_epoch\_acc, val\_epoch\_f1, f1\_score)

                    if metrics[0] is None:

                        print(f"Stopping validation at epoch {epoch + 1} due to invalid loss.")

                        break

                    loss, val\_epoch\_loss, val\_epoch\_acc, val\_epoch\_f1, total = metrics

                    val\_total\_preds += total

                val\_loss\_to\_track = val\_epoch\_loss / len(val\_dl)

                val\_acc\_to\_track = val\_epoch\_acc / val\_total\_preds

                val\_f1\_to\_track = val\_epoch\_f1 / len(val\_dl)

                val\_losses.append(val\_loss\_to\_track)

                val\_accs.append(val\_acc\_to\_track)

                val\_f1s.append(val\_f1\_to\_track.cpu().item() if torch.is\_tensor(val\_f1\_to\_track) else val\_f1\_to\_track)

                print(f"{epoch + 1}-epoch validation process is completed!")

                print(f"{epoch + 1}-epoch validation loss     -> {val\_loss\_to\_track:.3f}")

                print(f"{epoch + 1}-epoch validation accuracy -> {val\_acc\_to\_track:.3f}")

                print(f"{epoch + 1}-epoch validation f1-score -> {val\_f1\_to\_track:.3f}")

                # Save model based on validation accuracy

                if val\_acc\_to\_track > (best\_acc + threshold):

                    os.makedirs(save\_dir, exist\_ok=True)

                    best\_acc = val\_acc\_to\_track

                    torch.save(m.state\_dict(), f"{save\_dir}/{save\_prefix}\_best\_model.pth")

                    print(f"Model saved at epoch {epoch + 1} with validation accuracy: {val\_acc\_to\_track:.3f} to {save\_dir}/{save\_prefix}\_best\_model.pth")

                    not\_improved = 0

                else:

                    not\_improved += 1

                    print(f"Accuracy did not improve for {not\_improved} epochs")

                    if not\_improved >= patience:

                        print(f"Stopping training since accuracy did not improve for {patience} epochs.")

                        break

            # Step the learning rate scheduler based on validation loss

            scheduler.step(val\_loss\_to\_track)

            print(f"Completed epoch {epoch + 1}/{epochs}.")

        # Save metrics to a CSV file for plotting

        metrics\_df = pd.DataFrame({

            "Epoch": list(range(1, len(tr\_losses) + 1)),

            "Train Loss": tr\_losses,

            "Val Loss": val\_losses,

            "Train Accuracy": tr\_accs,

            "Val Accuracy": val\_accs,

            "Train F1": tr\_f1s,

            "Val F1": val\_f1s

        })

        metrics\_path = os.path.join(save\_dir, "training\_metrics.csv")

        metrics\_df.to\_csv(metrics\_path, index=False)

        print(f"Saved training metrics to {metrics\_path}")

    except Exception as e:

        print(f"An error occurred: {str(e)}")

        import traceback

        traceback.print\_exc()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**train\_oral\_disease.py:**

# train\_oral\_disease.py

# Suppress the huggingface\_hub symlink warning

import os

os.environ["HF\_HUB\_DISABLE\_SYMLINKS\_WARNING"] = "1"

import torch

import numpy as np

from glob import glob

from PIL import Image

from torch.utils.data import random\_split, Dataset, DataLoader

from torchvision import transforms as T

import timm

import torchmetrics

from tqdm import tqdm

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import confusion\_matrix

# Set random seed for reproducibility

torch.manual\_seed(2024)

# Step 1: Define CustomDataset and DataLoaders

class CustomDataset(Dataset):

    def \_\_init\_\_(self, root, transformations=None):

        self.transformations = transformations

        self.im\_paths = sorted(glob(f"{root}/\*/\*/\*.jpg"))

        self.cls\_names, self.cls\_counts, count = {}, {}, 0

        for idx, im\_path in enumerate(self.im\_paths):

            class\_name = self.get\_class(im\_path)

            if class\_name not in self.cls\_names:

                self.cls\_names[class\_name] = count

                self.cls\_counts[class\_name] = 1

                count += 1

            else:

                self.cls\_counts[class\_name] += 1

    def get\_class(self, path):

        # Extract only the class name without the full path

        return os.path.basename(os.path.dirname(path))

    def \_\_len\_\_(self):

        return len(self.im\_paths)

    def \_\_getitem\_\_(self, idx):

        im\_path = self.im\_paths[idx]

        try:

            im = Image.open(im\_path).convert("RGB")

        except Exception as e:

            print(f"Error loading image {im\_path}: {e}")

            return None, None

        gt = self.cls\_names[self.get\_class(im\_path)]

        if self.transformations is not None:

            im = self.transformations(im)

        return im, gt

def get\_dls(root, transformations, bs, split=[0.9, 0.05, 0.05], ns=4):

    ds = CustomDataset(root=root, transformations=transformations)

    total\_len = len(ds)

    tr\_len = int(total\_len \* split[0])

    vl\_len = int(total\_len \* split[1])

    ts\_len = total\_len - (tr\_len + vl\_len)

    tr\_ds, vl\_ds, ts\_ds = random\_split(dataset=ds, lengths=[tr\_len, vl\_len, ts\_len])

    tr\_dl = DataLoader(tr\_ds, batch\_size=bs, shuffle=True, num\_workers=ns)

    val\_dl = DataLoader(vl\_ds, batch\_size=bs, shuffle=False, num\_workers=ns)

    ts\_dl = DataLoader(ts\_ds, batch\_size=1, shuffle=False, num\_workers=ns)

    return tr\_dl, val\_dl, ts\_dl, ds.cls\_names

# Step 4: AI Model Training and Validation

def train\_setup(m, num\_classes):

    device = "cuda" if torch.cuda.is\_available() else "cpu"

    print(f"Using device: {device}")

    optimizer = torch.optim.Adam(params=m.parameters(), lr=3e-4, weight\_decay=1e-4)  # Added weight decay

    scheduler = torch.optim.lr\_scheduler.ReduceLROnPlateau(optimizer, mode='min', factor=0.1, patience=5, verbose=True)

    return (

        m.to(device).eval(),

        100,  # Increased to 100 epochs

        device,

        torch.nn.CrossEntropyLoss(),

        optimizer,

        scheduler

    )

def to\_device(batch, device):

    return batch[0].to(device), batch[1].to(device)

def get\_metrics(model, ims, gts, loss\_fn, epoch\_loss, epoch\_acc, epoch\_f1, f1\_score):

    preds = model(ims)

    loss = loss\_fn(preds, gts)

    if torch.isnan(loss) or torch.isinf(loss):

        print("Invalid loss detected!")

        return None, epoch\_loss, epoch\_acc, epoch\_f1

    pred\_labels = torch.argmax(preds, dim=1)

    return (

        loss,

        epoch\_loss + loss.item(),

        epoch\_acc + (pred\_labels == gts).sum().item(),

        epoch\_f1 + f1\_score(preds, gts),

        pred\_labels  # Return predictions for confusion matrix

    )

def main():

    try:

        # Clear GPU memory if using GPU

        if torch.cuda.is\_available():

            torch.cuda.empty\_cache()

        # Define the root path and transformations

        root = r"C:\Users\shafe\.cache\kagglehub\datasets\salmansajid05\oral-diseases\versions\3"

        mean, std, im\_size = [0.485, 0.456, 0.406], [0.229, 0.224, 0.225], 224

        train\_tfs = T.Compose([

            T.Resize((im\_size, im\_size)),

            T.RandomHorizontalFlip(p=0.5),

            T.RandomRotation(degrees=30),

            T.RandomAffine(degrees=0, scale=(0.8, 1.2), shear=10),

            T.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2),

            T.RandomApply([T.GaussianBlur(kernel\_size=3)], p=0.2),

            T.ToTensor(),

            T.Normalize(mean=mean, std=std)

        ])

        val\_test\_tfs = T.Compose([

            T.Resize((im\_size, im\_size)),

            T.ToTensor(),

            T.Normalize(mean=mean, std=std)

        ])

        # Get the data loaders

        tr\_dl, val\_dl, ts\_dl, classes = get\_dls(root=root, transformations=train\_tfs, bs=32)

        \_, val\_dl, ts\_dl, \_ = get\_dls(root=root, transformations=val\_test\_tfs, bs=32)

        # Print some information about the data loaders

        print(f"Number of batches in training DataLoader: {len(tr\_dl)}")

        print(f"Number of batches in validation DataLoader: {len(val\_dl)}")

        print(f"Number of batches in test DataLoader: {len(ts\_dl)}")

        print(f"Classes: {classes}")

        # Initialize the MobileNetV2 model with dropout

        m = timm.create\_model("mobilenetv2\_100", pretrained=True, num\_classes=len(classes), drop\_rate=0.3)

        # Training setup

        m, epochs, device, loss\_fn, optimizer, scheduler = train\_setup(m, len(classes))

        f1\_score = torchmetrics.F1Score(task="multiclass", num\_classes=len(classes)).to(device)

        save\_prefix, save\_dir = "oral\_disease", r"C:\Users\shafe\SmartDentAi-project"

        print("Start training...")

        # Training loop variables

        best\_acc, threshold, not\_improved, patience = 0.0, 0.01, 0, 10  # Track best accuracy

        tr\_losses, val\_losses, tr\_accs, val\_accs, tr\_f1s, val\_f1s = [], [], [], [], [], []

        for epoch in range(epochs):

            print(f"Starting epoch {epoch + 1}/{epochs}...")

            # Training phase

            epoch\_loss, epoch\_acc, epoch\_f1 = 0, 0, 0

            m.train()

            for idx, batch in tqdm(enumerate(tr\_dl), total=len(tr\_dl)):

                ims, gts = to\_device(batch, device)

                metrics = get\_metrics(m, ims, gts, loss\_fn, epoch\_loss, epoch\_acc, epoch\_f1, f1\_score)

                if metrics[0] is None:  # Invalid loss

                    print(f"Stopping epoch {epoch + 1} due to invalid loss.")

                    break

                loss, epoch\_loss, epoch\_acc, epoch\_f1, \_ = metrics

                optimizer.zero\_grad()

                loss.backward()

                torch.nn.utils.clip\_grad\_norm\_(m.parameters(), max\_norm=1.0)  # Gradient clipping

                optimizer.step()

            tr\_loss\_to\_track = epoch\_loss / len(tr\_dl)

            tr\_acc\_to\_track = epoch\_acc / len(tr\_dl.dataset)

            tr\_f1\_to\_track = epoch\_f1 / len(tr\_dl)

            tr\_losses.append(tr\_loss\_to\_track)

            tr\_accs.append(tr\_acc\_to\_track)

            tr\_f1s.append(tr\_f1\_to\_track.cpu().item() if torch.is\_tensor(tr\_f1\_to\_track) else tr\_f1\_to\_track)

            print(f"{epoch + 1}-epoch train process is completed!")

            print(f"{epoch + 1}-epoch train loss          -> {tr\_loss\_to\_track:.3f}")

            print(f"{epoch + 1}-epoch train accuracy      -> {tr\_acc\_to\_track:.3f}")

            print(f"{epoch + 1}-epoch train f1-score      -> {tr\_f1\_to\_track:.3f}")

            # Validation phase

            print(f"Starting validation for epoch {epoch + 1}...")

            m.eval()

            val\_epoch\_loss, val\_epoch\_acc, val\_epoch\_f1 = 0, 0, 0

            all\_preds, all\_gts = [], []

            with torch.no\_grad():

                for idx, batch in enumerate(val\_dl):

                    ims, gts = to\_device(batch, device)

                    metrics = get\_metrics(m, ims, gts, loss\_fn, val\_epoch\_loss, val\_epoch\_acc, val\_epoch\_f1, f1\_score)

                    if metrics[0] is None:  # Invalid loss

                        print(f"Stopping validation at epoch {epoch + 1} due to invalid loss.")

                        break

                    loss, val\_epoch\_loss, val\_epoch\_acc, val\_epoch\_f1, preds = metrics

                    all\_preds.extend(preds.cpu().numpy())

                    all\_gts.extend(gts.cpu().numpy())

            val\_loss\_to\_track = val\_epoch\_loss / len(val\_dl)

            val\_acc\_to\_track = val\_epoch\_acc / len(val\_dl.dataset)

            val\_f1\_to\_track = val\_epoch\_f1 / len(val\_dl)

            val\_losses.append(val\_loss\_to\_track)

            val\_accs.append(val\_acc\_to\_track)

            val\_f1s.append(val\_f1\_to\_track.cpu().item() if torch.is\_tensor(val\_f1\_to\_track) else val\_f1\_to\_track)

            print(f"{epoch + 1}-epoch validation process is completed!")

            print(f"{epoch + 1}-epoch validation loss     -> {val\_loss\_to\_track:.3f}")

            print(f"{epoch + 1}-epoch validation accuracy -> {val\_acc\_to\_track:.3f}")

            print(f"{epoch + 1}-epoch validation f1-score -> {val\_f1\_to\_track:.3f}")

            # Save model based on validation accuracy

            if val\_acc\_to\_track > (best\_acc + threshold):

                os.makedirs(save\_dir, exist\_ok=True)

                best\_acc = val\_acc\_to\_track

                torch.save(m.state\_dict(), f"{save\_dir}/{save\_prefix}\_best\_model.pth")

                print(f"Model saved at epoch {epoch + 1} with validation accuracy: {val\_acc\_to\_track:.3f} to {save\_dir}/{save\_prefix}\_best\_model.pth")

                not\_improved = 0

            else:

                not\_improved += 1

                print(f"Accuracy did not improve for {not\_improved} epochs")

                if not\_improved >= patience:

                    print(f"Stopping training since accuracy did not improve for {patience} epochs.")

                    break

            # Step the learning rate scheduler

            scheduler.step(val\_loss\_to\_track)

            print(f"Completed epoch {epoch + 1}/{epochs}.")

        # Save metrics to a CSV file

        metrics\_df = pd.DataFrame({

            "Epoch": list(range(1, len(tr\_losses) + 1)),

            "Train Loss": tr\_losses,

            "Val Loss": val\_losses,

            "Train Accuracy": tr\_accs,

            "Val Accuracy": val\_accs,

            "Train F1": tr\_f1s,

            "Val F1": val\_f1s

        })

        metrics\_path = os.path.join(save\_dir, "oral\_disease\_metrics.csv")

        metrics\_df.to\_csv(metrics\_path, index=False)

        print(f"Saved training metrics to {metrics\_path}")

        # Plot learning curves

        plt.figure(figsize=(15, 10))

        # Loss

        plt.subplot(3, 1, 1)

        plt.plot(metrics\_df["Epoch"], metrics\_df["Train Loss"], label="Train Loss", marker="o")

        plt.plot(metrics\_df["Epoch"], metrics\_df["Val Loss"], label="Val Loss", marker="o")

        plt.title("Training and Validation Loss")

        plt.xlabel("Epoch")

        plt.ylabel("Loss")

        plt.legend()

        plt.grid()

        # Accuracy

        plt.subplot(3, 1, 2)

        plt.plot(metrics\_df["Epoch"], metrics\_df["Train Accuracy"], label="Train Accuracy", marker="o")

        plt.plot(metrics\_df["Epoch"], metrics\_df["Val Accuracy"], label="Val Accuracy", marker="o")

        plt.title("Training and Validation Accuracy")

        plt.xlabel("Epoch")

        plt.ylabel("Accuracy")

        plt.legend()

        plt.grid()

        # F1-Score

        plt.subplot(3, 1, 3)

        plt.plot(metrics\_df["Epoch"], metrics\_df["Train F1"], label="Train F1", marker="o")

        plt.plot(metrics\_df["Epoch"], metrics\_df["Val F1"], label="Val F1", marker="o")

        plt.title("Training and Validation F1-Score")

        plt.xlabel("Epoch")

        plt.ylabel("F1-Score")

        plt.legend()

        plt.grid()

        plt.tight\_layout()

        plt.savefig(os.path.join(save\_dir, "oral\_disease\_learning\_curves.png"))

        plt.close()

        print(f"Saved learning curves to {save\_dir}/oral\_disease\_learning\_curves.png")

        # Plot confusion matrix for the validation set

        class\_names = list(classes.keys())

        cm = confusion\_matrix(all\_gts, all\_preds)

        plt.figure(figsize=(10, 8))

        sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class\_names, yticklabels=class\_names)

        plt.title("Confusion Matrix (Validation Set)")

        plt.xlabel("Predicted")

        plt.ylabel("True")

        plt.xticks(rotation=45, ha="right")

        plt.yticks(rotation=0)

        plt.tight\_layout()

        plt.savefig(os.path.join(save\_dir, "oral\_disease\_confusion\_matrix.png"))

        plt.close()

        print(f"Saved confusion matrix to {save\_dir}/oral\_disease\_confusion\_matrix.png")

    except Exception as e:

        print(f"An error occurred: {str(e)}")

        import traceback

        traceback.print\_exc()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Oral/templates:**

**index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Oral Disease Prediction</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">

    <style>

        .head{

            display: flex;

            justify-content: space-between;

        }

    </style>

</head>

<body class="bg-light">

    <div class="container mt-5">

        <div class="head">

            <h2 class="text-center">Oral Disease Prediction</h2>

            <button class="btn btn-primary" onclick="goBack(2)"> Sign Out </button>

        </div>

        <form action="/" method="post" enctype="multipart/form-data" class="mt-4">

            <div class="mb-3">

                <input type="file" name="file" class="form-control" required>

            </div>

            <button type="submit" class="btn btn-primary">Upload and Predict</button>

        </form>

        {% if prediction %}

        <div class="card mt-4">

            <div class="card-header text-center">

                <h4>Prediction Result</h4>

            </div>

            <div class="card-body">

                {% if img\_path %}

                <div class="text-center">

                    <img src="{{ img\_path }}" alt="Uploaded Image" class="img-thumbnail" width="200">

                </div>

                {% endif %}

                <table class="table table-bordered mt-3">

                    <thead class="table-dark">

                        <tr>

                            <th>Category</th>

                            <th>Details</th>

                        </tr>

                    </thead>

                    <tbody>

                        <tr>

                            <td><strong>Predicted Disease</strong></td>

                            <td>{{ prediction }}</td>

                        </tr>

                        <tr>

                            <td><strong>Symptoms</strong></td>

                            <td>{{ details.symptoms }}</td>

                        </tr>

                        <tr>

                            <td><strong>Causes</strong></td>

                            <td>{{ details.causes }}</td>

                        </tr>

                        <tr>

                            <td><strong>Treatment</strong></td>

                            <td>{{ details.treatment }}</td>

                        </tr>

                        <tr>

                            <td><strong>Precautions</strong></td>

                            <td>{{ details.precautions }}</td>

                        </tr>

                    </tbody>

                </table>

            </div>

        </div>

        {% endif %}

    </div>

    <script>

        function goBack() {

          window.history.back(-2);

      }

      </script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**Oral/app.py:**

**app.py:**

import torch

import torch.nn as nn

import torchvision.transforms as transforms

from torchvision import models

from PIL import Image

from flask import Flask, request, render\_template

import os

app = Flask(\_\_name\_\_, template\_folder="templates")

# Define the model class

class OralDiseaseModel(nn.Module):

    def \_\_init\_\_(self):

        super(OralDiseaseModel, self).\_\_init\_\_()

        self.model = models.mobilenet\_v2(weights=models.MobileNet\_V2\_Weights.DEFAULT)  # Ensure pretrained model matches training

        self.model.classifier[1] = nn.Linear(1280, 3)  # Adjust output layer to match 3 classes

    def forward(self, x):

        return self.model(x)

# Load the model

def load\_model():

    model = OralDiseaseModel()

    model.load\_state\_dict(torch.load("model\oral\_model.pth", map\_location=torch.device('cpu')), strict=False)  # Allow missing keys

    model.eval()

    return model

model = load\_model()

# Define class labels

class\_labels = {0: "Calculus", 1: "Gingivitis", 2: "Hypodontia"}

# Diagnosis information for each predicted class

diagnosis\_info = {

    "Calculus": {

        "symptoms": "Yellow or brown deposits on teeth, bad breath, gum irritation.",

        "causes": "Plaque hardens into tartar due to poor oral hygiene.",

        "treatment": "Professional dental cleaning, scaling, and root planing.",

        "precautions": "Regular brushing, flossing, and dental check-ups."

    },

    "Gingivitis": {

        "symptoms": "Red, swollen gums, bleeding while brushing or flossing.",

        "causes": "Plaque build-up causing gum inflammation.",

        "treatment": "Improved oral hygiene, professional cleaning, and antibacterial mouthwash.",

        "precautions": "Regular brushing, flossing, and avoiding tobacco products."

    },

    "Hypodontia": {

        "symptoms": "Missing one or more teeth from birth.",

        "causes": "Genetic factors, developmental disorders.",

        "treatment": "Orthodontic treatments, dental implants, or bridges.",

        "precautions": "Early diagnosis and regular dental consultations."

    }

}

# Define transformations

transform = transforms.Compose([

    transforms.Resize((224, 224)),

    transforms.ToTensor(),

    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])

])

@app.route('/', methods=['GET', 'POST'])

def index():

    prediction = None

    details = {}

    img\_path = None

    if request.method == 'POST':

        if 'file' not in request.files:

            return render\_template('index.html', prediction="No file uploaded", details=details, img\_path=img\_path)

        file = request.files['file']

        if file.filename == '':

            return render\_template('index.html', prediction="No file selected", details=details, img\_path=img\_path)

        # Save and process image

        file\_path = os.path.join('static', file.filename)

        file.save(file\_path)

        image = Image.open(file\_path).convert('RGB')  # Ensure image is in RGB mode

        image = transform(image).unsqueeze(0)

        # Predict

        with torch.no\_grad():

            outputs = model(image)

            \_, predicted = torch.max(outputs, 1)

            predicted\_class = class\_labels[predicted.item()]

            details = diagnosis\_info.get(predicted\_class, {})

            prediction = predicted\_class

            img\_path = file\_path

    return render\_template('index.html', prediction=prediction, details=details, img\_path=img\_path)

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True, port=5003)

**Oral/predict\_oral.ipynb:**

**predict\_oral.ipynb:**

import os

import numpy as np

import pandas as pd

import torch

import timm

from torchvision import transforms as T

from PIL import Image

import torch

from torch.utils.data import Dataset, DataLoader, random\_split

import matplotlib.pyplot as plt

from glob import glob

import random

# Suppress the huggingface\_hub symlink warning

os.environ["HF\_HUB\_DISABLE\_SYMLINKS\_WARNING"] = "1"

# Set random seed for reproducibility

torch.manual\_seed(2024)

# Custom Dataset class

class CustomDataset(Dataset):

    def \_\_init\_\_(self, root, transformations=None):

        self.transformations = transformations

        self.im\_paths = sorted(glob(f"{root}/\*/\*/\*.jpg"))

        print(f"Found {len(self.im\_paths)} images in dataset")

        if len(self.im\_paths) == 0:

            print(f"Warning: No images found in {root}/\*/\*/\*.jpg")

        else:

            print(f"Sample image paths: {self.im\_paths[:5]}")

        self.cls\_names, self.cls\_counts, count = {}, {}, 0

        for idx, im\_path in enumerate(self.im\_paths):

            class\_name = self.get\_class(im\_path)

            if class\_name not in self.cls\_names:

                self.cls\_names[class\_name] = count

                self.cls\_counts[class\_name] = 1

                count += 1

            else:

                self.cls\_counts[class\_name] += 1

        print(f"Classes found: {self.cls\_names}")

        print(f"Class counts: {self.cls\_counts}")

    def get\_class(self, path):

        return os.path.basename(os.path.dirname(path))

    def \_\_len\_\_(self):

        return len(self.im\_paths)

    def \_\_getitem\_\_(self, idx):

        im\_path = self.im\_paths[idx]

        try:

            im = Image.open(im\_path).convert("RGB")

        except Exception as e:

            print(f"Error loading image {im\_path}: {e}")

            return None, None, None

        gt = self.cls\_names[self.get\_class(im\_path)]

        if self.transformations is not None:

            im\_transformed = self.transformations(im)

        else:

            im\_transformed = T.ToTensor()(im)

        return im\_transformed, gt, im

# Custom collate function

def custom\_collate(batch):

    batch = [item for item in batch if item[0] is not None]

    if not batch:

        print("Warning: Entire batch is None after filtering")

        return None, None, None

    ims = torch.stack([item[0] for item in batch])

    gts = torch.tensor([item[1] for item in batch], dtype=torch.long)

    orig\_ims = [item[2] for item in batch]

    return ims, gts, orig\_ims

# Function to get DataLoaders with random\_split

def get\_dls(root, transformations, bs=32, split=[0.9, 0.05, 0.05], ns=0):

    ds = CustomDataset(root=root, transformations=transformations)

    total\_len = len(ds)

    tr\_len = int(total\_len \* split[0])

    vl\_len = int(total\_len \* split[1])

    ts\_len = total\_len - (tr\_len + vl\_len)

    tr\_ds, vl\_ds, ts\_ds = random\_split(dataset=ds, lengths=[tr\_len, vl\_len, ts\_len])

    tr\_dl = DataLoader(tr\_ds, batch\_size=bs, shuffle=True, num\_workers=ns, collate\_fn=custom\_collate)

    val\_dl = DataLoader(vl\_ds, batch\_size=bs, shuffle=False, num\_workers=ns, collate\_fn=custom\_collate)

    ts\_dl = DataLoader(ts\_ds, batch\_size=bs, shuffle=False, num\_workers=ns, collate\_fn=custom\_collate)

    return tr\_dl, val\_dl, ts\_dl, ds.cls\_names

# Define transformations

mean, std, im\_size = [0.485, 0.456, 0.406], [0.229, 0.224, 0.225], 224

train\_tfs = T.Compose([

    T.Resize((im\_size, im\_size)),

    T.RandomHorizontalFlip(p=0.5),

    T.RandomRotation(degrees=30),

    T.RandomAffine(degrees=0, scale=(0.8, 1.2), shear=10),

    T.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2),

    T.RandomApply([T.GaussianBlur(kernel\_size=3)], p=0.2),

    T.ToTensor(),

    T.Normalize(mean=mean, std=std)

])

val\_test\_tfs = T.Compose([

    T.Resize((im\_size, im\_size)),

    T.ToTensor(),

    T.Normalize(mean=mean, std=std)

])

# Function to convert tensor back to image for visualization

def tensor\_2\_im(t, t\_type="rgb"):

    gray\_tfs = T.Compose([

        T.Normalize(mean=[0.], std=[1/0.5]),

        T.Normalize(mean=[-0.5], std=[1])

    ])

    rgb\_tfs = T.Compose([

        T.Normalize(mean=[0., 0., 0.], std=[1/0.229, 1/0.224, 1/0.225]),

        T.Normalize(mean=[-0.485, -0.456, -0.406], std=[1., 1., 1.])

    ])

    invTrans = gray\_tfs if t\_type == "gray" else rgb\_tfs

    return (invTrans(t) \* 255).detach().squeeze().cpu().permute(1, 2, 0).numpy().astype(np.uint8) if t\_type == "gray" else (invTrans(t) \* 255).detach().cpu().permute(1, 2, 0).numpy().astype(np.uint8)

# Visualization function

def visualize(data, n\_ims, rows, cmap=None, cls\_names=None):

    assert cmap in ["rgb", "gray"], "Rasmni oq-qora yoki rangli ekanini aniqlashtirib bering!"

    if cmap == "rgb":

        cmap = "viridis"

    plt.figure(figsize=(20, 10))

    indekslar = [random.randint(0, len(data) - 1) for \_ in range(n\_ims)]

    for idx, indeks in enumerate(indekslar):

        im, gt, \_ = data[indeks]

        if im is None:

            continue

        plt.subplot(rows, n\_ims // rows, idx + 1)

        if cmap:

            plt.imshow(tensor\_2\_im(im, cmap), cmap=cmap)

        else:

            plt.imshow(tensor\_2\_im(im))

        plt.axis('off')

        if cls\_names is not None:

            plt.title(f"GT -> {cls\_names[int(gt)]}")

        else:

            plt.title(f"GT -> {gt}")

    plt.show()

# Load the trained PyTorch model (.pth)

pth\_model\_path = r"C:\Users\shafe\SmartDentAi-project\model\oral\_disease\_best\_model.pth"

# Initialize the model architecture (must match the one used during training)

model = timm.create\_model("mobilenetv2\_100", pretrained=False, num\_classes=3, drop\_rate=0.3)

# Load the state dict

model.load\_state\_dict(torch.load(pth\_model\_path))

print(f"Loaded PyTorch model from {pth\_model\_path}")

# Set device and move model to device

device = "cuda" if torch.cuda.is\_available() else "cpu"

model = model.to(device)

model.eval()  # Set to evaluation mode

print(f"Model moved to device: {device}")

# Dataset root path

root = r"C:\Users\shafe\.cache\kagglehub\datasets\salmansajid05\oral-diseases\versions\3"

# Get DataLoaders

tr\_dl, val\_dl, ts\_dl, classes = get\_dls(root, val\_test\_tfs, bs=32)

# Reverse the cls\_names dictionary

cls\_names\_reverse = {v: k for k, v in classes.items()}

# Function to perform predictions with confidence threshold

def predict\_and\_visualize(dataloader, split\_name, num\_samples=5, confidence\_threshold=0.1):

    predictions = []

    true\_labels = []

    original\_images = []

    confidences = []

    for batch in dataloader:

        if batch[0] is None:

            print(f"Skipping batch in {split\_name} split: batch[0] is None")

            continue

        ims, gts, orig\_ims = batch

        print(f"Processing batch in {split\_name} split: {ims.shape[0]} images")

        # Move inputs to device

        ims = ims.to(device)

        # Get predictions using PyTorch model

        with torch.no\_grad():

            preds = model(ims)

            pred\_probs = torch.softmax(preds, dim=1)  # Convert logits to probabilities

            pred\_probs = pred\_probs.cpu().numpy()

            pred\_labels = np.argmax(pred\_probs, axis=1)

            pred\_probs\_max = np.max(pred\_probs, axis=1)

        # Apply confidence threshold

        for i in range(len(pred\_labels)):

            if pred\_probs\_max[i] >= confidence\_threshold:

                predictions.append(pred\_labels[i])

                true\_labels.append(gts[i].item())

                original\_images.append(orig\_ims[i])

                confidences.append(pred\_probs\_max[i])

            else:

                print(f"Skipping prediction with low confidence: {pred\_probs\_max[i]:.4f}")

    predictions = np.array(predictions)

    true\_labels = np.array(true\_labels)

    print(f"Total predictions in {split\_name} split (after threshold): {len(predictions)}")

    print(f"Total true labels in {split\_name} split: {len(true\_labels)}")

    print(f"Total images for visualization: {len(original\_images)}")

    # Visualize sample images without confidence scores

    if len(original\_images) == 0:

        print(f"Warning: No images to visualize for {split\_name} split.")

    else:

        plt.figure(figsize=(15, num\_samples \* 3))

        for i in range(min(num\_samples, len(original\_images))):

            img = original\_images[i]

            pred = predictions[i]

            true = true\_labels[i]

            pred\_label = cls\_names\_reverse[pred]

            true\_label = cls\_names\_reverse[true]

            plt.subplot(num\_samples, 1, i + 1)

            plt.imshow(img)

            plt.axis('off')

            plt.title(f"Pred: {pred\_label}\nTrue: {true\_label}", fontsize=10)

        plt.tight\_layout()

        plt.show()

    return predictions, true\_labels

# Perform predictions with a confidence threshold of 0.1

print("Processing Test Split...")

test\_preds, test\_labels = predict\_and\_visualize(ts\_dl, "Test", num\_samples=5, confidence\_threshold=0.1)

print("\nProcessing Validation Split...")

val\_preds, val\_labels = predict\_and\_visualize(val\_dl, "Validation", num\_samples=5, confidence\_threshold=0.1)

# Save predictions to CSV

test\_df = pd.DataFrame({

    "Predicted": [cls\_names\_reverse[pred] for pred in test\_preds],

    "True": [cls\_names\_reverse[true] for true in test\_labels]

})

test\_df.to\_csv("oral\_disease\_test\_predictions.csv", index=False)

print("Test predictions saved to 'oral\_disease\_test\_predictions.csv'")

val\_df = pd.DataFrame({

    "Predicted": [cls\_names\_reverse[pred] for pred in val\_preds],

    "True": [cls\_names\_reverse[true] for true in val\_labels]

})

val\_df.to\_csv("oral\_disease\_val\_predictions.csv", index=False)

print("Validation predictions saved to 'oral\_disease\_val\_predictions.csv'")

# Visualize datasets

print("\nVisualizing Training Dataset...")

visualize(tr\_dl.dataset, 20, 4, "rgb", list(classes.keys()))

print("\nVisualizing Validation Dataset...")

visualize(val\_dl.dataset, 20, 4, "rgb", list(classes.keys()))

print("\nVisualizing Test Dataset...")

visualize(ts\_dl.dataset, 20, 4, "rgb", list(classes.keys()))

**ortho/code/static:**

**styles.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    box-sizing: border-box;

    margin: 0;

    padding: 0;

    scrollbar-width: none;

}

/\* form \*/

.predict{

    text-align: center;

    margin-top: 50px;

    background-color: #eae9e9;

    padding-top: 20px;

    padding-bottom: 40px;

    margin-left: 200px;

    margin-right: 200px;

    border-radius: 20px;

    height: 400px;

}

input[type="file"] {

    margin: 20px;

    background-color: #d3d3d3;

    width: 700px;

    margin-top: 50px;

}

input[type="submit"] {

    padding: 10px 20px;

    font-size: 16px;

}

header{

    /\* background-color: #31cab3; \*/

   background-image: linear-gradient(270deg, rgb(32, 209, 171), rgb(7, 84, 65));

    color: aliceblue;

    padding: 5px;

    box-shadow: 1px 1px 10px 2px #d9d8d8;

    height: 65px;

    border-radius: 50px 50px 50px 50px;

    padding-top: 10px;

    margin-top: 10px;

    margin-left: 150px;

    margin-right: 150px;

}

nav{

    display: flex;

    justify-content: space-around;

    gap: 30px;

    height: 55px;

    margin-top: 0;

}

/\* nav .head{

    margin-left: 100px;

}

nav .head1{

    margin-left: 750px;

} \*/

.form{

    background-color: #ffffff;

    padding: 1px;

    padding-bottom: 40px;

    border-radius:0px 0px 10px 10px ;

}

.h1-xray{

    color: #2c6b7a;

    margin-top: 40px;

}

.btn{

    background-color: #31cab3;

    border-color: #31cab3;

    border-radius: 10px;

    margin-top: 120px;

    color: white;

    width: 200px;

    padding: 10px;

    font-family: "Poppins", serif;

}

.img{

    background-color: #e8e8e8;

    border-radius: 10px;

    padding: 10px;

}

.head{

    display: flex;

    gap: 50px;

    margin: 0px;

    padding: 0;

}

.heads{

    display: flex;

    background-color:#31cab3;

    background-color: transparent;

    padding: 5px;

    border-radius: 5px;

    color: #e8e8e8;

}

.heads:hover{

    /\* background-color: #2cb5a0; \*/

    transform: rotate(5deg);

    transform: scale(1.1);

    color: #ffffff;

}

.pic{

    /\* display: flex; \*/

    justify-content: center;

    align-items: center;

    text-align: center;

    padding-top: 0px;

    gap: 60px;

    padding-bottom: 50px;

}

.title{

    color: #076b5c;

    font-family: "Poppins", serif;

    font-weight: 300;

    font-style: normal;

    margin-top: 30px;

}

.card-title{

    font-family: "Poppins", serif;

    color: #076b5c;

}

.head1{

    /\* margin-left: 50em; \*/

    background-color: transparent;

    margin-bottom: 0px;

    display: flex;

    padding: 0;

    gap: 50px;

}

.name{

    display: flex;

    /\* background-color:#31cab3; \*/

    background-color: transparent;

    color: #e8e8e8;

    font-size: 25px;

    margin-right: 0px;

    font-family: "Poppins", serif;

  font-weight: 400;

  font-style: normal;

    /\* margin-top: 10px; \*/

}

.cards{

    display: flex;

    gap: 50px;

    justify-content: center;

    margin-right: 30px;

    padding-bottom: 30px;

    margin-top: 50px;

}

.card-text{

    color: #b4b4b4;

}

/\* .cards-1{

    display: flex;

    gap: 50px;

    justify-content: center;

    margin-right: 30px;

} \*/

a{

    text-decoration: none;

    color: black;

}

footer{

    background-color: #ffffff;

    text-align: center;

    justify-content: center;

    align-items: center;

    margin-top: 100px;

}

.icons{

    width: 35px;

}

.main\_img{

    border-radius: 20px 20px 50px 50px;

    width: 1000px;

    /\* border-radius: 86% 79% 89% 51% / 82% 39% 100% 86% ; \*/

}

.span-title{

    color: #44ff00;

    font-style: italic;

}

.span-subtitle{

    color: #b0adad;

    font-weight: 0;

}

.main-title{

    font-size: 50px;

    font-family:Georgia, 'Times New Roman', Times, serif;

  background: -webkit-linear-gradient(45deg,#39a696,rgb(0, 255, 136), #027459,  rgb(5, 5, 5));

  -webkit-background-clip: text;

  -webkit-text-fill-color: transparent;

}

.span-cards{

    color: #31cab3;

}

.span-dent{

    color: #0b7a87;

}

.appoint-logo{

    padding-bottom: 18px;

}

.btn.btn-outline-primary{

    background: transparent;

    border-color: white;

    margin-top:0;

}

.main-logo{

    color: #66c9b2;

    font-weight: 400;

}

.head2{

    color: #cacbcb;

    font-style: italic;

    font-style: "Poppins", serif;

    font-size: 25px;

    font-weight: bold;

}

.messsageDiv{

    background-color: hsl(327,90%,28%);

    color: white;

    padding: 10px 20px;

    margin: 10px;

    border-radius: 5px;

    font-size: 1rem;

    opacity: 0;

    animation: fadeOut 7s forwards;

}

.user\_profile{

    width: 45px;

    height: 45px;

}

**ortho/code/templates:**

**index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Image Processing</title>

    <link rel="icon" href="{{ url\_for('static', filename='assets/favicon.ico') }}" type="image/x-icon">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/styles.css">

    <style>

        body { font-family: Arial, sans-serif; text-align: center; }

        .imgg { max-width: 100%; height: auto; margin-top: 20px; border: 2px solid black; }

        .image-container { display: flex; justify-content: center; gap: 20px; margin-top: 20px; }

        .image-box { text-align: center; }

        .main{

            background-color: rgb(237, 237, 237);

            margin: 50px;

            margin-left: 150px;

            margin-right: 150px;

            border-radius: 50px;

        }

        .matter{

            color: rgb(17, 117, 107);

            font-weight: 600;

            font-style: italic;

            font-family: 'Poppins', sans-serif;

            padding: 20px;

        }

        .title{

            display: flex;

            justify-content: space-around;

            align-items: center;

        }

        .btn.btn-primary{

            margin: 0;

        }

    </style>

</head>

<body>

    <header class="header">

        <!-- <h1>Upload X-ray Image for Disease Prediction</h1> -->

        <nav>

            <ul class="head2">

              <a><li class="heads">Smart<span class="main-logo">Dent</span>Ai</li></a>

            </ul>

            <ul class="head">

                <!-- <a href="" class="" style="text-decoration: none;"><li class="heads">Home</li></a> -->

                <a href="" class="" style="text-decoration: none;"><li class="heads"><img class="icons" src="../static/assets/homee.png" alt="Image"></li></a>

                <a href="" class="" style="text-decoration: none;"><li class="heads"><img class="icons" src="../static/assets/calll.png" alt="Image"></li></a>

                <a href="" class="" style="text-decoration: none;"><li class="heads"><img class="icons" src="../static/assets/about\_uss.png" alt="Image"></li></a>

                <!-- <a href="" class="" style="text-decoration: none;"><li class="heads">Login</li></a>

                <a href="" class="" style="text-decoration: none;"><li class="heads">Logout</li></a> -->

            </ul>

            <ul class="head1">

                <!-- <a href="" style="text-decoration: none;"><li class="name">Smart<span class="span-dent">Dent</span>Ai</li></a> -->

                <a href=""><img class="user\_profile" src="../static//assets//man.png" alt=""></a>

                <button class="btn btn-outline-primary" id="logout">Sign out</button>

            </ul>

        </nav>

    </header>

    <div class="main">

        <div class="title">

            <h2 class="matter">Upload an Image for Processing</h2>

            <!-- <button class="btn btn-primary" action="/home" method="post">Back</button> -->

                <button class="btn btn-primary" >Back</button>

        </div>

    <form method="post" enctype="multipart/form-data">

        <input type="file" name="file" accept="image/\*" required>

        <button type="submit" class="btn btn-primary">Upload</button>

    </form>

    {% if input\_image and output\_image %}

        <div class="image-container">

            <div class="image-box">

                <h3>Input Image</h3>

                <img class="imgg" src="{{ input\_image }}" alt="Input Image">

            </div>

            <div class="image-box">

                <h3>Processed Output</h3>

                <img class="imgg" src="{{ output\_image }}" alt="Output Image">

            </div>

        </div>

        <br>

        <a href="{{ output\_image }}" download>Download Processed Image</a>

    {% endif %}

    </div>

</body>

</html>

**ortho/code:**

**app.py:**

import os

import torch

import torchvision.transforms as transforms

from PIL import Image

from flask import Flask, request, render\_template, send\_file

from model import Generator

# Initialize Flask app

app = Flask(\_\_name\_\_)

UPLOAD\_FOLDER = "static/uploads"

OUTPUT\_FOLDER = "static/outputs"

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

os.makedirs(OUTPUT\_FOLDER, exist\_ok=True)

# Load model

checkpoint\_path = r"C:\MajorProject\SmartDentAi-project\ortho\code\checkpoints\model\_epoch\_195.pth"  # Update this path

device = torch.device('cuda' if torch.cuda.is\_available() else 'cpu')

generator = Generator().to(device)

checkpoint = torch.load(checkpoint\_path, map\_location=device)

generator.load\_state\_dict(checkpoint['generator\_state\_dict'])

generator.eval()

# Image transformation

transform = transforms.Compose([

    transforms.Resize((256, 256)),

    transforms.ToTensor(),

    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])

])

def process\_image(image\_path, output\_path):

    """Process uploaded image through the generator."""

    input\_image = Image.open(image\_path).convert('RGB')

    input\_tensor = transform(input\_image).unsqueeze(0).to(device)

    with torch.no\_grad():

        output\_tensor = generator(input\_tensor)

    output\_tensor = output\_tensor \* 0.5 + 0.5

    output\_tensor = output\_tensor.clamp(0, 1).cpu().squeeze(0)

    output\_image = transforms.ToPILImage()(output\_tensor)

    output\_image = output\_image.resize((300, 300), Image.LANCZOS)

    output\_image.save(output\_path)

@app.route("/", methods=["GET", "POST"])

def index():

    input\_file\_path = None

    output\_file\_path = None

    if request.method == "POST":

        if "file" not in request.files:

            return "No file uploaded", 400

        file = request.files["file"]

        if file.filename == "":

            return "No selected file", 400

        input\_file\_path = os.path.join(UPLOAD\_FOLDER, file.filename)

        output\_file\_path = os.path.join(OUTPUT\_FOLDER, f"processed\_{file.filename}")

        file.save(input\_file\_path)

        process\_image(input\_file\_path, output\_file\_path)

        return render\_template("index.html", input\_image=input\_file\_path, output\_image=output\_file\_path)

    return render\_template("index.html", input\_image=None, output\_image=None)

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, port=5006)

**ortho/code:**

**model.py:**# model.py

import torch

import torch.nn as nn

import torch.nn.functional as F

from torchvision import models

from torch.utils.data import Dataset

from PIL import Image

import os

class ResidualBlock(nn.Module):

    def \_\_init\_\_(self, channels):

        super(ResidualBlock, self).\_\_init\_\_()

        self.block = nn.Sequential(

            nn.Conv2d(channels, channels, 3, padding=1),

            nn.InstanceNorm2d(channels),

            nn.ReLU(inplace=True),

            nn.Conv2d(channels, channels, 3, padding=1),

            nn.InstanceNorm2d(channels)

        )

    def forward(self, x):

        return x + self.block(x)

class Generator(nn.Module):

    def \_\_init\_\_(self):

        super(Generator, self).\_\_init\_\_()

        # Initial convolution block

        self.init\_block = nn.Sequential(

            nn.Conv2d(3, 64, 7, padding=3, padding\_mode='reflect'),

            nn.InstanceNorm2d(64),

            nn.ReLU(inplace=True)

        )

        # Downsampling

        self.down1 = nn.Sequential(

            nn.Conv2d(64, 128, 3, stride=2, padding=1),

            nn.InstanceNorm2d(128),

            nn.ReLU(inplace=True)

        )

        self.down2 = nn.Sequential(

            nn.Conv2d(128, 256, 3, stride=2, padding=1),

            nn.InstanceNorm2d(256),

            nn.ReLU(inplace=True)

        )

        # Residual blocks

        self.res\_blocks = nn.Sequential(

            \*[ResidualBlock(256) for \_ in range(6)]

        )

        # Upsampling

        self.up1 = nn.Sequential(

            nn.ConvTranspose2d(256, 128, 3, stride=2, padding=1, output\_padding=1),

            nn.InstanceNorm2d(128),

            nn.ReLU(inplace=True)

        )

        self.up2 = nn.Sequential(

            nn.ConvTranspose2d(128, 64, 3, stride=2, padding=1, output\_padding=1),

            nn.InstanceNorm2d(64),

            nn.ReLU(inplace=True)

        )

        # Output layer

        self.output\_block = nn.Sequential(

            nn.Conv2d(64, 3, 7, padding=3, padding\_mode='reflect'),

            nn.Tanh()

        )

    def forward(self, x):

        x = self.init\_block(x)

        x = self.down1(x)

        x = self.down2(x)

        x = self.res\_blocks(x)

        x = self.up1(x)

        x = self.up2(x)

        x = self.output\_block(x)

        return x

class Discriminator(nn.Module):

    def \_\_init\_\_(self):

        super(Discriminator, self).\_\_init\_\_()

        def discriminator\_block(in\_filters, out\_filters, normalize=True, stride=2):

            layers = [nn.Conv2d(in\_filters, out\_filters, 4, stride=stride, padding=1)]

            if normalize:

                layers.append(nn.InstanceNorm2d(out\_filters))

            layers.append(nn.LeakyReLU(0.2, inplace=True))

            return layers

        self.model = nn.Sequential(

            \*discriminator\_block(3, 64, normalize=False),  # 128x128

            \*discriminator\_block(64, 128),                 # 64x64

            \*discriminator\_block(128, 256),               # 32x32

            \*discriminator\_block(256, 512),               # 16x16

            nn.Conv2d(512, 1, 4, stride=1, padding=1)    # 15x15

        )

    def forward(self, x):

        return self.model(x)  # Returns a tensor of shape (batch\_size, 1, 15, 15)

class TeethDataset(Dataset):

    def \_\_init\_\_(self, input\_dir, target\_dir, transform=None):

        self.input\_dir = input\_dir

        self.target\_dir = target\_dir

        self.transform = transform

        self.image\_names = [f for f in os.listdir(input\_dir) if f.endswith(('.png', '.jpg', '.jpeg'))]

    def \_\_len\_\_(self):

        return len(self.image\_names)

    def \_\_getitem\_\_(self, idx):

        img\_name = self.image\_names[idx]

        input\_path = os.path.join(self.input\_dir, img\_name)

        target\_path = os.path.join(self.target\_dir, img\_name)

        input\_image = Image.open(input\_path).convert('RGB')

        target\_image = Image.open(target\_path).convert('RGB')

        if self.transform:

            input\_image = self.transform(input\_image)

            target\_image = self.transform(target\_image)

        return input\_image, target\_image

**train.py:**

# train.py

import os

import torch

import torch.nn as nn

from torch.utils.data import DataLoader

import torchvision.transforms as transforms

from torch.cuda.amp import autocast, GradScaler

from model import Generator, Discriminator, TeethDataset

import torchvision.models as models

class PerceptualLoss(nn.Module):

    def \_\_init\_\_(self):

        super(PerceptualLoss, self).\_\_init\_\_()

        vgg = models.vgg16(pretrained=True).features[:23].eval()

        for param in vgg.parameters():

            param.requires\_grad = False

        self.vgg = vgg

        self.criterion = nn.L1Loss()

    def forward(self, x, y):

        vgg\_x = self.vgg(x)

        vgg\_y = self.vgg(y)

        return self.criterion(vgg\_x, vgg\_y)

def train\_model(generator, discriminator, train\_loader, num\_epochs, device):

    # Optimizers

    g\_optimizer = torch.optim.Adam(generator.parameters(), lr=0.0002, betas=(0.5, 0.999))

    d\_optimizer = torch.optim.Adam(discriminator.parameters(), lr=0.0002, betas=(0.5, 0.999))

    # Loss functions

    criterion\_gan = nn.BCEWithLogitsLoss()

    criterion\_pixel = nn.L1Loss()

    criterion\_perceptual = PerceptualLoss().to(device)

    # Gradient scaler

    scaler\_g = GradScaler()

    scaler\_d = GradScaler()

    for epoch in range(num\_epochs):

        for i, (misaligned, aligned) in enumerate(train\_loader):

            batch\_size = misaligned.size(0)

            # Move data to device

            real\_imgs = aligned.to(device)

            misaligned\_imgs = misaligned.to(device)

            # Create labels matching discriminator output size (15x15)

            valid = torch.ones((batch\_size, 1, 15, 15), requires\_grad=False).to(device)

            fake = torch.zeros((batch\_size, 1, 15, 15), requires\_grad=False).to(device)

            # -----------------

            #  Train Generator

            # -----------------

            g\_optimizer.zero\_grad()

            with autocast():

                # Generate a batch of images

                gen\_imgs = generator(misaligned\_imgs)

                # Adversarial loss

                disc\_out = discriminator(gen\_imgs)

                g\_loss\_adv = criterion\_gan(disc\_out, valid)

                # Pixel-wise loss

                g\_loss\_pixel = criterion\_pixel(gen\_imgs, real\_imgs)

                # Perceptual loss

                g\_loss\_perceptual = criterion\_perceptual(gen\_imgs, real\_imgs)

                # Total loss

                g\_loss = g\_loss\_adv + 100 \* g\_loss\_pixel + 10 \* g\_loss\_perceptual

            scaler\_g.scale(g\_loss).backward()

            scaler\_g.step(g\_optimizer)

            scaler\_g.update()

            # -----------------

            #  Train Discriminator

            # -----------------

            d\_optimizer.zero\_grad()

            with autocast():

                # Real loss

                pred\_real = discriminator(real\_imgs)

                loss\_real = criterion\_gan(pred\_real, valid)

                # Fake loss

                pred\_fake = discriminator(gen\_imgs.detach())

                loss\_fake = criterion\_gan(pred\_fake, fake)

                # Total discriminator loss

                d\_loss = (loss\_real + loss\_fake) / 2

            scaler\_d.scale(d\_loss).backward()

            scaler\_d.step(d\_optimizer)

            scaler\_d.update()

            # Print progress

            if i % 100 == 0:

                print(f"[Epoch {epoch}/{num\_epochs}] "

                      f"[Batch {i}/{len(train\_loader)}] "

                      f"[D loss: {d\_loss.item():.4f}] "

                      f"[G loss: {g\_loss.item():.4f}]")

        # Save models periodically

        if (epoch + 1) % 5 == 0:

            save\_path = f'checkpoints/model\_epoch\_{epoch+1}.pth'

            torch.save({

                'epoch': epoch,

                'generator\_state\_dict': generator.state\_dict(),

                'discriminator\_state\_dict': discriminator.state\_dict(),

                'g\_optimizer\_state\_dict': g\_optimizer.state\_dict(),

                'd\_optimizer\_state\_dict': d\_optimizer.state\_dict(),

            }, save\_path)

            print(f"Model saved to {save\_path}")

def main():

    # Set device

    device = torch.device('cuda' if torch.cuda.is\_available() else 'cpu')

    print(f"Using device: {device}")

    # Create checkpoints directory

    os.makedirs('checkpoints', exist\_ok=True)

    # Data paths

    input\_dir = r"C:\MajorProject\SmartDentAi-project\ortho\data\train\input"

    target\_dir = r"C:\MajorProject\SmartDentAi-project\ortho\data\train\target"

    # Transform

    transform = transforms.Compose([

        transforms.Resize((256, 256)),

        transforms.ToTensor(),

        transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])

    ])

    # Create dataset and dataloader

    dataset = TeethDataset(input\_dir, target\_dir, transform)

    dataloader = DataLoader(

        dataset,

        batch\_size=4,

        shuffle=True,

        num\_workers=2,

        pin\_memory=True

    )

    # Initialize models

    generator = Generator().to(device)

    discriminator = Discriminator().to(device)

    if torch.cuda.device\_count() > 1:

        print(f"Using {torch.cuda.device\_count()} GPUs")

        generator = nn.DataParallel(generator)

        discriminator = nn.DataParallel(discriminator)

    # Train model

    train\_model(generator, discriminator, dataloader, num\_epochs=200, device=device)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Remote/static:**

**styles.css:**

body {

    background: #6a11cb;

    background: -webkit-linear-gradient(to right, rgba(106, 17, 203, 1), rgba(37, 117, 252, 1));

    background: linear-gradient(to right, rgba(106, 17, 203, 1), rgba(37, 117, 252, 1));

}

#root {

            width: 100vw;

            height: 100vh;

        }

        /\* Remove the outline from the video embed \*/

        #root video {

            outline: none;

        }

        /\* Style the "Back to Dashboard" button \*/

        .back-to-dashboard {

            position: absolute;

            bottom: 20px;

            left: 20px;

        }

        .back-to-dashboard button {

            padding: 10px 20px;

            background-color: #ffffff;

            border: none;

            border-radius: 5px;

            cursor: pointer;

        }

        /\* Hide any content that overflows the viewport \*/

        body {

            overflow: hidden;

        }

**Remote/templates:**

**base.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>{% block title %} My Website {% endblock %}</title>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

  <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

</head>

<body>

<section class="vh-100 gradient-custom">

  <div class="container py-5 h-100">

    <div class="row d-flex justify-content-center align-items-center h-100">

      <div class="col-12 col-md-8 col-lg-6 col-xl-5">

        <div class="card bg-dark text-white" style="border-radius: 1rem;">

          <div class="card-body p-5 text-center">

              {% block content %}{% endblock %}

          </div>

        </div>

      </div>

    </div>

  </div>

</section>

</body>

</html>

**dashboard.html:**

{% extends 'base.html' %}

{% block title %}My Dashboard{% endblock %}

{% block content %}

    <h1><b>Welcome</b> {{ first\_name }} {{ last\_name }}</h1>

    <hr>

<!-- "Create or Join Meeting" button -->

    <div class="create-meeting-button">

        <div class="button-container">

            <a class="btn btn-primary" href="{{ url\_for('meeting') }}">Create Meeting</a>

            <a class="btn btn-primary" href="{{ url\_for('join') }}">Join Meeting</a>

            <a class="btn btn-outline-light" href="{{ url\_for('logout') }}">Logout</a>

        </div>

{% endblock %}

**join.html:**

{% extends 'base.html' %}

{% block title %}Join Meeting{% endblock %}

{% block content %}

<div class="container">

        <h1 class="text-center mt-5">Join Meeting</h1>

        <form method="POST" action="{{ url\_for('join') }}" class="mt-4">

            <div class="form-group">

                <input type="text" placeholder="Room ID:" class="form-control" id="roomID" name="roomID" required>

            </div>

            <button type="submit" class="btn btn-primary btn-lg">Join</button>

        </form>

    </div>

{% endblock %}

**login.html:**

{% extends 'base.html' %}

{% block title %}Login{% endblock %}

{% block content %}

{% with messages = get\_flashed\_messages() %}

    {% if messages %}

            <div class="alert alert-success">

                {{ messages | join('<br>') | safe }}

            </div>

    {% endif %}

{% endwith %}

<div class="mb-md-5 mt-md-4 pb-5">

    <h2 class="fw-bold mb-2 text-uppercase">Login</h2>

    <p class="text-white-50 mb-5">Please enter your login and password</p>

    <form action="{{ url\_for('login') }}" method="POST">

    {{ form.csrf\_token }}

    {{ form.hidden\_tag() }}

    <div class="form-outline form-white mb-4">

    {{ form.email.label }} {{ form.email(class="form-control form-control-lg", id="email") }}<br>

    {{ form.password.label }} {{ form.password(class="form-control form-control-lg", id="password") }}

    </div>

    <button class="btn btn-outline-light btn-lg px-5" type="submit">Login</button>

</form>

</div>

<div><p class="mb-0">Don't have an account? <a href="{{ url\_for('register') }}" class="text-white-50 fw-bold">Register</a></p> </div>

{% endblock %}

**meeting.html:**

<html>

<head>

    <title>Meetings</title>

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

</head>

<body>

    <div id="root"></div>

    <div class="back-to-dashboard">

        <a href="{{ url\_for('dashboard') }}" style="text-decoration: none;">

            <button>Back to Dashboard</button>

        </a>

    </div>

</body>

<script src="https://unpkg.com/@zegocloud/zego-uikit-prebuilt/zego-uikit-prebuilt.js"></script>

<script>

window.onload = function () {

    function getUrlParams(url) {

        let urlStr = url.split('?')[1];

        const urlSearchParams = new URLSearchParams(urlStr);

        const result = Object.fromEntries(urlSearchParams.entries());

        return result;

    }

        // Generate a Token by calling a method.

        // @param 1: appID

        // @param 2: serverSecret

        // @param 3: Room ID

        // @param 4: User ID

        // @param 5: Username

    const roomID = getUrlParams(window.location.href)['roomID'] || (Math.floor(Math.random() \* 10000) + "");

    const userID = Math.floor(Math.random() \* 10000) + "";

    const userName = "{{username}}";

    const appID = appID;

    const serverSecret = "serverSecret";

    const kitToken = ZegoUIKitPrebuilt.generateKitTokenForTest(appID, serverSecret, roomID, userID, userName);

        const zp = ZegoUIKitPrebuilt.create(kitToken);

        zp.joinRoom({

            container: document.querySelector("#root"),

            sharedLinks: [{

                name: 'Personal link',

                url: window.location.protocol + '//' + window.location.host  + window.location.pathname + '?roomID=' + roomID,

            }],

            scenario: {

                mode: ZegoUIKitPrebuilt.VideoConference,

            },

            turnOnMicrophoneWhenJoining: true,

            turnOnCameraWhenJoining: true,

            showMyCameraToggleButton: true,

            showMyMicrophoneToggleButton: true,

            showAudioVideoSettingsButton: true,

            showScreenSharingButton: true,

            showTextChat: true,

            showUserList: true,

            maxUsers: 2,

            layout: "Auto",

            showLayoutButton: false,

            });

}

</script>

</html>

**register.html:**

{% extends 'base.html' %}

{% block title %}Register{% endblock %}

{% block content %}

<div class="mb-md-5 mt-md-4 pb-5">

    <form action="{{ url\_for('register') }}" method="POST">

    {{ form.csrf\_token }}

    {{ form.hidden\_tag() }}

    <h2 class="fw-bold mb-2 text-uppercase">Register</h2>

    <p class="text-white-50 mb-5">Create an Account</p>

    <div class="form-outline form-white mb-4">

        {{ form.first\_name.label }} {{ form.first\_name(class="form-control form-control-lg", id="firstname") }}<br>

        {{ form.last\_name.label }} {{ form.last\_name(class="form-control form-control-lg", id="lastname") }}<br>

        {{ form.email.label }} {{ form.email(class="form-control form-control-lg", id="email") }}<br>

        {{ form.username.label }} {{ form.username(class="form-control form-control-lg", id="username") }}<br>

        {{ form.password.label }} {{ form.password(class="form-control form-control-lg", id="password") }}

    </div>

    <button class="btn btn-outline-light btn-lg px-5" type="submit">Register</button>

</form>

</div>

<div><p class="mb-0">Already have an account? <a href="{{ url\_for('login') }}" class="text-white-50 fw-bold">Login</a></p> </div>

{% endblock %}

**Remote/.gitignore:**

# Byte-compiled / optimized / DLL files

\_\_pycache\_\_/

\*.py[cod]

\*$py.class

# C extensions

\*.so

# Distribution / packaging

.Python

build/

develop-eggs/

dist/

downloads/

eggs/

.eggs/

lib/

lib64/

parts/

sdist/

var/

wheels/

share/python-wheels/

\*.egg-info/

.installed.cfg

\*.egg

MANIFEST

# PyInstaller

#  Usually these files are written by a python script from a template

#  before PyInstaller builds the exe, so as to inject date/other infos into it.

\*.manifest

\*.spec

# Installer logs

pip-log.txt

pip-delete-this-directory.txt

# Unit test / coverage reports

htmlcov/

.tox/

.nox/

.coverage

.coverage.\*

.cache

nosetests.xml

coverage.xml

\*.cover

\*.py,cover

.hypothesis/

.pytest\_cache/

cover/

# Translations

\*.mo

\*.pot

# Django stuff:

\*.log

local\_settings.py

db.sqlite3

db.sqlite3-journal

# Flask stuff:

instance/

.webassets-cache

# Scrapy stuff:

.scrapy

# Sphinx documentation

docs/\_build/

# PyBuilder

.pybuilder/

target/

# Jupyter Notebook

.ipynb\_checkpoints

# IPython

profile\_default/

ipython\_config.py

# pyenv

#   For a library or package, you might want to ignore these files since the code is

#   intended to run in multiple environments; otherwise, check them in:

# .python-version

# pipenv

#   According to pypa/pipenv#598, it is recommended to include Pipfile.lock in version control.

#   However, in case of collaboration, if having platform-specific dependencies or dependencies

#   having no cross-platform support, pipenv may install dependencies that don't work, or not

#   install all needed dependencies.

#Pipfile.lock

# poetry

#   Similar to Pipfile.lock, it is generally recommended to include poetry.lock in version control.

#   This is especially recommended for binary packages to ensure reproducibility, and is more

#   commonly ignored for libraries.

#   https://python-poetry.org/docs/basic-usage/#commit-your-poetrylock-file-to-version-control

#poetry.lock

# pdm

#   Similar to Pipfile.lock, it is generally recommended to include pdm.lock in version control.

#pdm.lock

#   pdm stores project-wide configurations in .pdm.toml, but it is recommended to not include it

#   in version control.

#   https://pdm.fming.dev/#use-with-ide

.pdm.toml

# PEP 582; used by e.g. github.com/David-OConnor/pyflow and github.com/pdm-project/pdm

\_\_pypackages\_\_/

# Celery stuff

celerybeat-schedule

celerybeat.pid

# SageMath parsed files

\*.sage.py

# Environments

.env

.venv

env/

venv/

ENV/

env.bak/

venv.bak/

# Spyder project settings

.spyderproject

.spyproject

# Rope project settings

.ropeproject

# mkdocs documentation

/site

# mypy

.mypy\_cache/

.dmypy.json

dmypy.json

# Pyre type checker

.pyre/

# pytype static type analyzer

.pytype/

# Cython debug symbols

cython\_debug/

# PyCharm

#  JetBrains specific template is maintained in a separate JetBrains.gitignore that can

#  be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore

#  and can be added to the global gitignore or merged into this file.  For a more nuclear

#  option (not recommended) you can uncomment the following to ignore the entire idea folder.

#.idea/

**Remote/main.py:**

**main.py:**

from flask import Flask, render\_template, request, redirect, url\_for, flash

from flask\_wtf import FlaskForm

from wtforms import StringField, PasswordField, EmailField

from wtforms.validators import DataRequired, Length

from flask\_sqlalchemy import SQLAlchemy

from flask\_login import login\_user, LoginManager, login\_required, current\_user, logout\_user

db = SQLAlchemy()

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = "my-secrets"

app.config['SQLALCHEMY\_DATABASE\_URI'] = "sqlite:///video-meeting.db"

db.init\_app(app)

login\_manager = LoginManager()

login\_manager.login\_view = "login"

login\_manager.init\_app(app)

@login\_manager.user\_loader

def load\_user(user\_id):

    return Register.query.get(int(user\_id))

class Register(db.Model):

    id = db.Column(db.Integer, primary\_key=True)

    email = db.Column(db.String(50), unique=True, nullable=False)

    first\_name = db.Column(db.String(50), nullable=False)

    last\_name = db.Column(db.String(50), nullable=False)

    username = db.Column(db.String(20), unique=True, nullable=False)

    password = db.Column(db.String(128), nullable=False)

    def is\_active(self):

        return True

    def get\_id(self):

        return str(self.id)

    def is\_authenticated(self):

        return True

with app.app\_context():

    db.create\_all()

class RegistrationForm(FlaskForm):

    email = EmailField(label='Email', validators=[DataRequired()])

    first\_name = StringField(label="First Name", validators=[DataRequired()])

    last\_name = StringField(label="Last Name", validators=[DataRequired()])

    username = StringField(label="Username", validators=[DataRequired(), Length(min=4, max=20)])

    password = PasswordField(label="Password", validators=[DataRequired(), Length(min=8, max=20)])

class LoginForm(FlaskForm):

    email = EmailField(label='Email', validators=[DataRequired()])

    password = PasswordField(label="Password", validators=[DataRequired()])

@app.route("/dashboard")

def home():

    return redirect(url\_for("dashboard"))

@app.route("/login", methods=["POST", "GET"])

def login():

    form = LoginForm()

    if request.method == "POST" and form.validate\_on\_submit():

        email = form.email.data

        password = form.password.data

        user = Register.query.filter\_by(email=email, password=password).first()

        if user:

            login\_user(user)

            return redirect(url\_for("dashboard"))

    return render\_template("login.html", form=form)

@app.route("/logout", methods=["GET"])

@login\_required

def logout():

    logout\_user()

    flash("You have been logged out successfully!", "info")

    return redirect(url\_for("login"))

@app.route("/register", methods=["POST", "GET"])

def register():

    form = RegistrationForm()

    if request.method == "POST" and form.validate\_on\_submit():

        new\_user = Register(

            email=form.email.data,

            first\_name=form.first\_name.data,

            last\_name=form.last\_name.data,

            username=form.username.data,

            password=form.password.data

        )

        db.session.add(new\_user)

        db.session.commit()

        flash("Account created Successfully! <br>You can now log in.", "success")

        return redirect(url\_for("login"))

    return render\_template("register.html", form=form)

@app.route("/")

@login\_required

def dashboard():

    return render\_template("dashboard.html", first\_name=current\_user.first\_name, last\_name=current\_user.last\_name)

@app.route("/meeting")

@login\_required

def meeting():

    return render\_template("meeting.html", username=current\_user.username)

@app.route("/join", methods=["GET", "POST"])

@login\_required

def join():

    if request.method == "POST":

        room\_id = request.form.get("roomID")

        return redirect(f"/meeting?roomID={room\_id}")

    return render\_template("join.html")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, port=5002)

**Remote/README.md:**

**README.md:**

**# Video Meeting Web App with Python Flask and ZegoCloud API**

Install with Terminal

- pip install Flask

- pip install Flask-WTF

- pip install Flask-SQLAlchemy

Get ZegoCloud API Keys from: https://bit.ly/3PNjNTW

Howto: https://youtu.be/n1VmVtMNmHc

**static/about\_us.css:**

**about\_us.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    box-sizing: border-box;

    margin: 0;

    padding: 0;

    scrollbar-width: none;

    width: 100%;

}

.top{

    display: flex;

    height: 200px;

    /\* background-image: linear-gradient(270deg, rgb(32, 209, 171), rgb(7, 84, 65)); \*/

    background: linear-gradient(rgba(0, 255, 183, 0.5), rgba(43, 223, 251, 0.5)),

    url('../static/assets/dashboard.jpeg');

    background-size: cover;

    background-position: center;

}

.top-left{

    background-color: transparent;

    text-align: center;

    margin: auto;

    font-family: "Poppins", serif;

    color: white;

}

.p-text{

    font-family: "Poppins",serif , monospace;

}

**static/chatbot.css:**

**chatbot.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

/\* chatbot/chatbot.css \*/

.chatbot-icon {

    position: fixed;

    bottom: 20px;

    right: 20px;

    cursor: pointer;

    z-index: 1000;

}

.chatbot-window {

    position: fixed;

    bottom: 70px; /\* Adjust based on icon size \*/

    right: 20px;

    width: 500px;

    max-height: 800px;

    background: #ffffff;

    border-radius: 10px;

    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);

    display: none; /\* Initially hidden \*/

    z-index: 1000;

    overflow: hidden;

    flex-direction: column;

}

.chatbot-header {

    /\* background: #31cab3; \*/

    background-image: linear-gradient(to right, black, #31cab3);

    color: white;

    padding: 10px;

    border-top-left-radius: 10px;

    border-top-right-radius: 10px;

    display: flex;

    justify-content: space-between;

    align-items: center;

    font-family: "Poppins", serif;

}

.bot-msg{

    color: #ffffff;

    font-family:'Times New Roman';

    font-size: 20px;

    padding-left: 20px;

    padding-top: 10px;

}

.chat-box {

    padding: 10px;

    flex-grow: 1;

    overflow-y: auto;

    max-height: 300px; /\* Limit height \*/

}

.user-message {

    background-color: #e0f7fa;

    align-self: flex-end;

    padding: 10px;

    border-radius: 10px;

    margin: 5px 0;

    max-width: 70%;

}

.bot-message {

    background-color: #f1f0f0;

    align-self: flex-start;

    padding: 10px;

    border-radius: 10px;

    margin: 5px 0;

    max-width: 70%;

}

textarea {

    resize: none;

    margin-left: 10px;

}

.btn {

    margin: 10px;

}

.msg-close-btn{

    background-color: hsl(0, 0%, 0%);

    color: #e0f7fa;

    border-color: #000000;

    border-radius: 20px;

    width: 40px;

}

**static/contact\_us.css;**

**contact\_us.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    box-sizing: border-box;

    margin: 0;

    padding: 0;

    scrollbar-width: none;

    width: 100%;

}

.top{

    display: flex;

    height: 200px;

    /\* background-image: linear-gradient(270deg, rgb(32, 209, 171), rgb(7, 84, 65)); \*/

    /\* background: linear-gradient(rgba(0, 255, 183, 0.5), rgba(43, 223, 251, 0.5)),  \*/

    background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),

    url('../static/assets/bgggg.jpg');

    background-size: cover;

    background-position: center;

}

.top-left{

    background-color: transparent;

    text-align: center;

    margin: auto;

    font-family: "Poppins", serif;

    color: white;

}

.p-text{

    font-family: "Poppins",serif , monospace;

}

.img-fluid.rounded-start{

    width: 300px;

}

.card.mb-3{

    margin: auto;

    margin-top: 50px;

    width: 48%;

    /\* border: none; \*/

    background-color: rgb(255, 255, 255);

    border-color: rgb(216, 216, 216);

}

.ul-icons{

    display: flex;

    list-style: none;

    padding: 0px;

    /\* margin-top: 30px; \*/

}

img{

    width: 50px;

}

.card-title{

    font-size: 50px;

    font-family: "Poppins", serif;

}

.col-md-4{

    width: 250px;

}

.card-text{

    bottom: auto;

}

.my-name{

    color: rgb(13, 154, 135);

}

.card-body{

    margin: auto 0;

}

.col-md-8{

    margin: auto 0;

}

.card-text{

    margin-top: 78px;

}

.position{

    color: rgb(154, 153, 153);

}

.info{

    color: rgb(12, 116, 98);

}

#ul-1{

    display: flex;

    gap: 5px;

}

.phone-mail{

    width: 50px;

}

.phone-no{

    display: flex;

    background-color: rgb(255, 255, 255);

    align-items: center;

    text-align: center;

}

.phone-text{

    margin: 0 10px;

}

#phone{

    width: 60%;

}

.email-id{

    display: flex;

    background-color: rgb(255, 255, 255);

    align-items: center;

    text-align: center;

}

.email-text{

    margin: 0 10px;

}

**static/dentalimg.css:**

**dentalimg.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    margin: 0;

    padding: 0;

    scrollbar-width: none;

}

/\* body {

    background-image: url("../static/assets/ortho.jpg");

    background-size: cover;

    background-repeat: no-repeat;

    -webkit-mask-image: linear-gradient(to right, transparent, rgb(0, 0, 0) 75%);

    mask-image: linear-gradient(to right, transparent, rgb(22, 1, 1) 75%);

} \*/

body{

    font-family: "Poppins", serif;

    font-style: italic;

}

.container-fluid{

    display: flex;

}

.ortho-name{

    color: rgb(4, 127, 132);

    font-size: 30px;

    font-weight: bolder;

}

.d-inline-block.align-text-top{

    display: inline-block;

}

.bg-body-tertiary {

    --bs-bg-opacity: 1;

    background-color: antiquewhite;

    /\* background-color: rgba(var(--bs-tertiary-bg-rgb), var(--bs-bg-opacity)) !important; \*/

}

.main{

    display: flex;

    width: 100%;

    height: 70vh;

}

.left{

    width: 38%;

    margin: 5px;

    margin-top: 10px;

}

.left .img-left{

    width: 600px;

    margin: auto;

    text-align: center;

    margin-top: 50px;

    margin-left: 25px;

}

.right{

    background-color: rgb(248, 248, 248);

    width: 63%;

    margin-top: 20px;

    border-radius: 50px;

    margin-right: 20px;

}

.title-name{

    font-size: 30px;

    text-align: center;

    margin-top: 70px;

}

.title-span{

    color: rgb(104, 181, 249);

}

.or-p{

    text-align: center;

}

.img{

    background-color: #e8e8e8;

    border-radius: 10px;

    padding: 10px;

}

/\* form \*/

.predict{

    text-align: center;

    margin-top: 100px;

    background-color: #eae9e9;

    padding-top: 20px;

    padding-bottom: 40px;

    margin-left: 200px;

    margin-right: 200px;

    border-radius: 20px;

}

input[type="file"] {

    margin: 20px;

    background-color: #d3d3d3;

    width: 500px;

}

input[type="submit"] {

    padding: 10px 20px;

    font-size: 16px;

}

footer{

    background-color: #ffffff;

    text-align: center;

    justify-content: center;

    align-items: center;

    margin-top: 100px;

    margin-bottom: 0px;

}

**static/index.js:**

**index.js:**

// Import the functions you need from the SDKs you need

import { initializeApp } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-analytics.js";

import { getAuth, signOut } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-auth.js";

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

    apiKey: "AIzaSyAs5EsYG\_4LSLOEZhJvEfS1Y6z0crmneO4",

    authDomain: "smartdentai.firebaseapp.com",

    projectId: "smartdentai",

    // storageBucket: "smartdentai.firebasestorage.app",

    storageBucket: "smartdentai.appspot.com",

    messagingSenderId: "182043265305",

    appId: "1:182043265305:web:47c60bfb1927b40928dd88",

    measurementId: "G-3ZK0LWZZLQ"

  };

  // Initialize Firebase

  const app = initializeApp(firebaseConfig);

  const analytics = getAnalytics(app);

  const auth = getAuth();

//Sign out function

const logoutUser = () => {

    signOut(auth).then(() => {

        // Sign-out successful.

        console.log("User signed out successfully");

        alert("signed out successfully!")

        window.location.href = "/";

    }).catch((error) => {

    // An error happened.

    console.log("Error signing out: ", error);

});

};

// Call the logout function when the button is clicked

document.getElementById("logout").addEventListener("click", logoutUser);

**static/login.css:**

**login.css:**

/\* From Uiverse.io by JohnnyCSilva \*/

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  background-color: white;

}

.text-overlay{

  position: absolute;

  top: 10%;

  left: 50%;

  transform: translate(-50%, -50%); /\* Centers the text \*/

  color: white; /\* Text color \*/

  font-size: 3rem; /\* Adjust font size \*/

  font-family: 'Poppins', serif;

  text-align: center;

  background: rgba(0, 0, 0, 0.5); /\* Optional background for better readability \*/

  padding: 10px 20px;

  border-radius: 10px; /\* Optional rounded corners \*/

}

.text-overlay-p{

  position: absolute;

  top: 15%;

  left: 50%;

  transform: translate(-50%, -50%); /\* Centers the text \*/

  color: rgb(197, 197, 197); /\* Text color \*/

  font-size: 1rem; /\* Adjust font size \*/

  font-family: 'Poppins', serif;

  text-align: center;

  background: rgba(0, 0, 0, 0.5); /\* Optional background for better readability \*/

  padding: 10px 20px;

  border-radius: 10px; /\* Optional rounded corners \*/

}

.login-img{

  width: 100%;

  height: 100%;

  object-fit: cover;    /\* Ensures the image covers the container \*/

  display: block;       /\* Removes default spacing from inline images \*/

  border: none;         /\* Removes any default border \*/

}

.main{

  display: flex;

  width: 100%;

  height:625px;

}

.left{

  position: relative;

  width: 58%;

  height: 100vh;

  background-color: #2d79f3;

  margin: 0;

  padding: 0;

}

body{

    background-color: #f0f0f0;

}

.form {

    display: flex;

    flex-direction: column;

    margin: 0 auto;

    margin-top: 5%;

    gap: 10px;

    background-color: #ffffff;

    padding: 30px;

    width: 30%;

    border-radius: 20px;

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

  }

  ::placeholder {

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

  }

  .form button {

    align-self: flex-end;

  }

  .flex-column > label {

    color: #151717;

    font-weight: 600;

  }

  .inputForm {

    border: 1.5px solid #ecedec;

    border-radius: 10px;

    height: 50px;

    display: flex;

    align-items: center;

    padding-left: 10px;

    transition: 0.2s ease-in-out;

  }

  .input {

    margin-left: 10px;

    border-radius: 10px;

    border: none;

    width: 100%;

    height: 100%;

  }

  .input:focus {

    outline: none;

  }

  .inputForm:focus-within {

    border: 1.5px solid #2d79f3;

  }

  .flex-row {

    display: flex;

    flex-direction: row;

    align-items: center;

    gap: 10px;

    justify-content: space-between;

  }

  .flex-row > div > label {

    font-size: 14px;

    color: black;

    font-weight: 400;

  }

  .span {

    font-size: 14px;

    margin-left: 5px;

    color: #2d79f3;

    font-weight: 500;

    cursor: pointer;

  }

  .button-submit {

    margin: 20px 0 10px 0;

    background-color: #151717;

    border: none;

    color: white;

    font-size: 15px;

    font-weight: 500;

    border-radius: 10px;

    height: 50px;

    width: 100%;

    cursor: pointer;

  }

  .p {

    text-align: center;

    color: black;

    font-size: 14px;

    margin: 5px 0;

  }

  .btn {

    margin-top: 10px;

    width: 100%;

    height: 50px;

    border-radius: 10px;

    display: flex;

    justify-content: center;

    align-items: center;

    font-weight: 500;

    gap: 10px;

    border: 1px solid #ededef;

    background-color: white;

    cursor: pointer;

    transition: 0.2s ease-in-out;

  }

  .btn:hover {

    border: 1px solid #2d79f3;

    ;

  }

  .name-login{

    text-align: center;

    color: rgb(156, 154, 154);

  }

  .span-name-login{

    color: rgb(30, 175, 124);

  }

**static/login.js:**

**login.js:**

// Import the functions you need from the SDKs you need

import { initializeApp } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-analytics.js";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

import { getAuth, sendPasswordResetEmail, signInWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-auth.js";

import { getFirestore, setDoc, doc } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-firestore.js";

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

  apiKey: "AIzaSyAs5EsYG\_4LSLOEZhJvEfS1Y6z0crmneO4",

  authDomain: "smartdentai.firebaseapp.com",

  projectId: "smartdentai",

  storageBucket: "smartdentai.firebasestorage.app",

  messagingSenderId: "182043265305",

  appId: "1:182043265305:web:47c60bfb1927b40928dd88",

  measurementId: "G-3ZK0LWZZLQ"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

const auth = getAuth(app);

const db = getFirestore();

function showMessage(message, divId) {

  var messageDiv = document.getElementById(divId);

  if (!messageDiv) return; // Prevents errors if the element doesn't exist

  messageDiv.style.display = "block";

  messageDiv.innerHTML = message;

  messageDiv.style.opacity = 1;

  setTimeout(() => {

    messageDiv.style.opacity = 0;

  }, 5000);

}

//submit

const signIn = document.getElementById('submit');

signIn.addEventListener('click', function(event){

    event.preventDefault()

    //Inputs

    const email = document.getElementById('email').value;

    const password = document.getElementById('password').value;

    signInWithEmailAndPassword(auth, email, password)

  .then((userCredential) => {

    // Signed up

    showMessage('Login is Successfull', 'signInMesssage');

    const user = userCredential.user;

    localStorage.setItem('LoggedId', user.uid);

    alert("Loggin in...")

    window.location.href = "/home";

    // ...

})

  .catch((error) => {

    const errorCode = error.code;

    const errorMessage = error.message;

    alert(errorMessage);

    // ..

    if(errorCode=='auth/invalid-credential'){

      showMessage('Invalid Email or Password', 'signInMesssage');

    }

    else{

      showMessage('Account does not Exist', 'signInMessage');

    }

  });

})

//reset

const reset = document.getElementById("reset");

reset.addEventListener('click', function(event){

  event.preventDefault()

  const email = document.getElementById('email').value;

  sendPasswordResetEmail(auth, email)

  .then(() => {

    // Password reset email sent!

    // ..

    alert("Email Sent")

  })

  .catch((error) => {

    const errorCode = error.code;

    const errorMessage = error.message;

    // ..

    alert(errorMessage);

  });

});

**static/main.css:**

**main.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

    \*{

        margin: 0;

        padding: 0;

        scrollbar-width: none;

    }

    /\* background-image: url('../static/assets/bgggg.jpg');  \*/

    body {

        background-image: url('../static/assets/bbg.jpg');

        /\* background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),

        url('../static/assets/bgggg.jpg'); \*/

        background-size: 100%;

        font-family: "Poppins", serif;

    }

    /\* ul {

        background-color: rgb(161, 158, 158);

        opacity: 75%;

        margin-left: 40% ;

        margin-right: 40%;

        border-radius: 10px;

        margin-top: 10px;

    } \*/

    .dash-img{

        width: 100%;

    }

    .nav-link.active {

        text-decoration: none;

        color: rgb(174, 174, 174);

    }

    .nav-link {

        text-decoration: none;

        color: rgb(174, 174, 174);

    }

    .nav-link:hover{

        color: rgb(96, 96, 96);

    }

    .nav-link.active:hover{

        color: rgb(96, 96, 96);

    }

    .d-grid.gap-2.col-6.mx-auto{

        position: absolute;

        bottom: 50px;

        right: 0px;

        left: 0px;

        justify-content: center;

        margin-left: 50%;

    }

    .btn.btn-primary{

        background-color: rgb(96, 96, 96);

        border-color: rgb(96, 96, 96);

        color: rgb(204, 204, 204);

        width: 200px;

    }

    .btn.btn-primary:hover{

        background-color: rgb(66, 65, 65);

        border-color: rgb(66, 65, 65);

    }

    .name-title{

        text-align: center;

        color: rgb(88, 106, 136);

        bottom: 0 auto;

    }

    .span-name-title{

        font-size: 50px;

        color: rgb(51, 86, 78);

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**static/orthodontics.css:**

**orthodontic.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    margin: 0;

    padding: 0;

    scrollbar-width: none;

}

/\* body {

    background-image: url("../static/assets/ortho.jpg");

    background-size: cover;

    background-repeat: no-repeat;

    -webkit-mask-image: linear-gradient(to right, transparent, rgb(0, 0, 0) 75%);

    mask-image: linear-gradient(to right, transparent, rgb(22, 1, 1) 75%);

} \*/

body{

    font-family: "Poppins", serif;

    font-style: italic;

}

.container-fluid{

    display: flex;

}

.ortho-name{

    color: rgb(4, 127, 132);

    font-size: 30px;

    font-weight: bolder;

}

.d-inline-block.align-text-top{

    display: inline-block;

}

.bg-body-tertiary {

    --bs-bg-opacity: 1;

    background-color: antiquewhite;

    /\* background-color: rgba(var(--bs-tertiary-bg-rgb), var(--bs-bg-opacity)) !important; \*/

}

.main{

    display: flex;

    width: 100%;

}

.left{

    width: 30%;

    margin: 5px;

    margin-top: 10px;

}

.left .img-left{

    width: 500px;

    margin: auto;

    text-align: center;

}

.right{

    background-color: rgb(248, 248, 248);

    width: 70%;

    margin: 5px;

}

.title-name{

    font-size: 30px;

    text-align: center;

    margin-top: 10px;

}

.title-span{

    color: rgb(104, 181, 249);

}

.or-p{

    text-align: center;

}

.img{

    background-color: #e8e8e8;

    border-radius: 10px;

    padding: 10px;

}

/\* form \*/

.predict{

    text-align: center;

    margin-top: 50px;

    background-color: #eae9e9;

    padding-top: 20px;

    padding-bottom: 40px;

    margin-left: 200px;

    margin-right: 200px;

    border-radius: 20px;

}

input[type="file"] {

    margin: 20px;

    background-color: #d3d3d3;

    width: 500px;

}

input[type="submit"] {

    padding: 10px 20px;

    font-size: 16px;

}

footer{

    background-color: #ffffff;

    text-align: center;

    justify-content: center;

    align-items: center;

    margin-top: 100px;

    margin-bottom: 0px;

}

**static/predict.css:**

**predict.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    box-sizing: border-box;

    margin: 0;

    padding: 0;

    scrollbar-width: none;

}

body{

    font-family: poppins, sans-serif;

}

h1, h2 {

    text-align: center;

    margin-top: 30px;

}

img {

    display: block;

    margin: 20px auto;

    width: 300px;

    height: auto;

}

table {

    width: 75%;

    margin: 20px auto;

    height: 300px;

    /\* border-collapse: collapse; \*/

}

table, th, td {

    border: 1px solid rgb(194, 193, 193);

    background-color: aliceblue;

}

th, td {

    padding: 10px;

    text-align: left;

}

th {

    background-color: #ffffff;

}

a{

    text-decoration: none;

    color: black;

}

.upload-img{

    border-radius: 20px;

}

.head2{

    margin-left: 320px;

}

.h2-predict{

    color: rgb(7, 126, 126);

}

.h1-predict{

    color: rgb(6, 128, 128);

}

.span-h1-predict{

    color: rgb(218, 25, 25);

}

.table-title{

    font-size: 20px;

    color: rgb(36, 114, 133);

}

.row-title{

    color: rgb(7, 119, 114);

}

td{

    color: rgb(129, 131, 131);

}

.icons-predict{

    width: 35px;

    margin: 0;

}

.head2-predict{

    color: #cacbcb;

    font-style: italic;

    font-style: "Poppins", serif;

    font-size: 25px;

    font-weight: bold;

    margin: 0;

    padding: 0;

}

.head-predict{

    display: flex;

    gap: 50px;

    margin: 0px;

    padding: 0;

}

.nav-predict{

    display: flex;

    justify-content: space-around;

    gap: 30px;

    height: 55px;

    margin-top: 0;

}

**static/profile.css:**

**profile.css:**

body {

    background-color: #f9f9fa

}

.padding {

    padding: 3rem !important

}

.user-card-full {

    overflow: hidden;

}

.card {

    border-radius: 5px;

    -webkit-box-shadow: 0 1px 20px 0 rgba(69,90,100,0.08);

    box-shadow: 0 1px 20px 0 rgba(69,90,100,0.08);

    border: none;

    margin-bottom: 30px;

}

.m-r-0 {

    margin-right: 0px;

}

.m-l-0 {

    margin-left: 0px;

}

.user-card-full .user-profile {

    border-radius: 5px 0 0 5px;

}

.bg-c-lite-green {

        background: -webkit-gradient(linear, left top, right top, from(#f29263), to(#ee5a6f));

    background: linear-gradient(to right, #ee5a6f, #f29263);

}

.user-profile {

    padding: 20px 0;

}

.card-block {

    padding: 1.25rem;

}

.m-b-25 {

    margin-bottom: 25px;

}

.img-radius {

    border-radius: 5px;

}

h6 {

    font-size: 14px;

}

.card .card-block p {

    line-height: 25px;

}

@media only screen and (min-width: 1400px){

p {

    font-size: 14px;

}

}

.card-block {

    padding: 1.25rem;

}

.b-b-default {

    border-bottom: 1px solid #e0e0e0;

}

.m-b-20 {

    margin-bottom: 20px;

}

.p-b-5 {

    padding-bottom: 5px !important;

}

.card .card-block p {

    line-height: 25px;

}

.m-b-10 {

    margin-bottom: 10px;

}

.text-muted {

    color: #919aa3 !important;

}

.b-b-default {

    border-bottom: 1px solid #e0e0e0;

}

.f-w-600 {

    font-weight: 600;

}

.m-b-20 {

    margin-bottom: 20px;

}

.m-t-40 {

    margin-top: 20px;

}

.p-b-5 {

    padding-bottom: 5px !important;

}

.m-b-10 {

    margin-bottom: 10px;

}

.m-t-40 {

    margin-top: 20px;

}

.user-card-full .social-link li {

    display: inline-block;

}

.user-card-full .social-link li a {

    font-size: 20px;

    margin: 0 10px 0 0;

    -webkit-transition: all 0.3s ease-in-out;

    transition: all 0.3s ease-in-out;

}

**static/profile.js:**

**profile.js:**

import { initializeApp } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-app.js";

import { getAuth, onAuthStateChanged, signOut, updateProfile } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-auth.js";

import { getFirestore, doc, getDoc } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-firestore.js";

import { getStorage, ref, uploadBytes, getDownloadURL } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-storage.js";

const firebaseConfig = {

  apiKey: "AIzaSyAs5EsYG\_4LSLOEZhJvEfS1Y6z0crmneO4",

  authDomain: "smartdentai.firebaseapp.com",

  projectId: "smartdentai",

  storageBucket: "smartdentai.firebasestorage.app",

  messagingSenderId: "182043265305",

  appId: "1:182043265305:web:47c60bfb1927b40928dd88",

  measurementId: "G-3ZK0LWZZLQ"

};

// Initialization

const app = initializeApp(firebaseConfig);

const auth = getAuth(app);

const db = getFirestore(app);

const storage = getStorage(app);

// Updated DOM elements to match profile.html

const userName = document.getElementById('username');

const userEmail = document.getElementById('email');

const profilePicture = document.getElementById('userProfilePicture');

const signOutButton = document.getElementById('logout');

const profilePictureInput = document.getElementById('profilePictureInput');

const uploadProfilePictureButton = document.getElementById('uploadProfilePicture');

onAuthStateChanged(auth, (user) => {

  console.log(user);

  if (user) {

    const userDocRef = doc(db, 'users', user.uid);

    getDoc(userDocRef)

      .then((docSnapshot) => {

        if (docSnapshot.exists()) {

          const userData = docSnapshot.data();

          userName.textContent = userData.name || "N/A";

          userEmail.textContent = userData.email || user.email;

        } else {

          userName.textContent = user.displayName || "N/A";

          userEmail.textContent = user.email || "N/A";

          alert("No user data found in Firestore. Displaying basic info.");

        }

        if (user.photoURL) {

          profilePicture.src = user.photoURL;

        }

      })

      .catch((error) => {

        console.error("Error fetching user data", error);

        userName.textContent = user.displayName || "N/A";

        userEmail.textContent = user.email || "N/A";

        if (user.photoURL) {

          profilePicture.src = user.photoURL;

        }

        alert("Failed to fetch user data from Firestore.");

      });

    // Handle profile picture upload

    if (uploadProfilePictureButton && profilePictureInput) {

      uploadProfilePictureButton.addEventListener('click', () => {

        const file = profilePictureInput.files[0];

        if (!file) {

          alert("Please select an image to upload.");

          return;

        }

        // Validate file type

        if (!file.type.startsWith('image/')) {

          alert("Please upload an image file.");

          return;

        }

        // Validate file size (max 5MB)

        const maxSizeInBytes = 5 \* 1024 \* 1024; // 5MB

        if (file.size > maxSizeInBytes) {

          alert("File is too large. Please upload an image smaller than 5MB.");

          return;

        }

        const storageRef = ref(storage, `profile-pictures/${user.uid}/${file.name}`);

        uploadBytes(storageRef, file)

          .then((snapshot) => {

            return getDownloadURL(snapshot.ref);

          })

          .then((downloadURL) => {

            // Refresh the user's token to ensure updateProfile succeeds

            return auth.currentUser.getIdToken(true).then(() => {

              return updateProfile(user, { photoURL: downloadURL }).then(() => downloadURL);

            });

          })

          .then((downloadURL) => {

            profilePicture.src = downloadURL;

            alert("Profile picture uploaded successfully!");

          })

          .catch((error) => {

            console.error("Error uploading profile picture:", error);

            alert(`Failed to upload profile picture: ${error.message}`);

          });

      });

    }

  } else {

    window.location.href = '/login';

  }

});

// Sign-out functionality

signOutButton.addEventListener('click', (event) => {

  event.preventDefault();

  signOut(auth)

    .then(() => {

      console.log("User signed out successfully.");

      window.location.href = '/login';

    })

    .catch((error) => {

      console.error("Sign-out error:", error);

      alert("Failed to sign out. Please try again.");

    });

});

**static/register.js:**

**register.js:**

// Import the functions you need from the SDKs you need

import { initializeApp } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-analytics.js";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

import { getAuth, createUserWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-auth.js";

import { getFirestore, setDoc, doc } from "https://www.gstatic.com/firebasejs/11.3.1/firebase-firestore.js";

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

  apiKey: "AIzaSyAs5EsYG\_4LSLOEZhJvEfS1Y6z0crmneO4",

  authDomain: "smartdentai.firebaseapp.com",

  projectId: "smartdentai",

  storageBucket: "smartdentai.firebasestorage.app",

  messagingSenderId: "182043265305",

  appId: "1:182043265305:web:47c60bfb1927b40928dd88",

  measurementId: "G-3ZK0LWZZLQ"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

const auth = getAuth(app);

const db = getFirestore();

function showMessage(message, divId) {

  var messageDiv = document.getElementById(divId);

  messageDiv.style.display = "block";

  messageDiv.innerHTML = message;

  messageDiv.style.opacity = 1;

  setTimeout(function() {

    messageDiv.style.opacity = 0;

  }, 5000);

}

//submit

const submit = document.getElementById('submit');

submit.addEventListener('click', function(event) {

  event.preventDefault();

  //Inputs

  const name = document.getElementById('name').value; // Added name input

  const email = document.getElementById('email').value;

  const password = document.getElementById('password').value;

  createUserWithEmailAndPassword(auth, email, password)

    .then((userCredential) => {

      // Signed up

      const user = userCredential.user;

      const userData = {

        name: name, // Store the name

        email: email,

        // Removed password for security

        createdAt: new Date().toISOString() // Optional: Add timestamp

      };

      showMessage('Account Created Successfully!', 'signUpMessage');

      const docRef = doc(db, "users", user.uid);

      setDoc(docRef, userData)

        .then(() => {

          window.location.href = "/login";

        })

        .catch((error) => {

          console.error("Error writing document", error);

        });

    })

    .catch((error) => {

      const errorCode = error.code;

      if (errorCode == 'auth/email-already-in-use') {

        showMessage('Email Already In Use!', 'signUpMessage');

      } else {

        showMessage('Unable Creating Account!', 'signUpMessage');

      }

    });

});

**static/scripts.js:**

**scripts.js:**

// function toggleChatbot() {

//     const chatbotWindow = document.getElementById("chatbot-window");

//     chatbotWindow.style.display = (chatbotWindow.style.display === "none" || chatbotWindow.style.display === "") ? "block" : "none";

// }

// const chatBox = document.querySelector(".chat-box");

// const messageInput = document.querySelector("#message-input");

// const sendBtn = document.querySelector("#send-btn");

// function addMessage(message, isUserMessage) {

//     const messageDiv = document.createElement("div");

//     messageDiv.classList.add("mt-3", "p-3", "rounded");

//     if (isUserMessage) {

//         messageDiv.classList.add("user-message");

//     } else {

//         messageDiv.classList.add("bot-message");

//     }

//     messageDiv.innerHTML = `

//         <p>${message}</p>

//     `;

//     chatBox.appendChild(messageDiv);

//     chatBox.scrollTop = chatBox.scrollHeight;

// }

// function sendMessage() {

//     const message = messageInput.value.trim();

//     if (message !== "") {

//         addMessage(message, true);

//         messageInput.value = "";  // Clear the input

//         fetch("/api", {

//             method: "POST",

//             headers: {

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

//             },

//             body: JSON.stringify({ message })

//         })

//         .then(response => response.json())

//         .then(data => {

//             if (data.response) {

//                 addMessage(data.response, false);

//             } else {

//                 addMessage("Error: " + data.error, false);

//             }

//         })

//         .catch(error => {

//             console.error(error);

//             addMessage("Error: " + error.message, false);

//         });

//     }

// }

// sendBtn.addEventListener("click", sendMessage);

// messageInput.addEventListener("keydown", event => {

//     if (event.keyCode === 13 && !event.shiftKey) {

//         event.preventDefault();

//         sendMessage();

//     }

// });

function toggleChatbot() {

    var chatbotWindow = document.getElementById("chatbot-window");

    if(!chatbotWindow) return;

    chatbotWindow.style.display = chatbotWindow.style.display === "none" ? "block" : "none";

}

document.getElementById("send-btn").addEventListener("click", async function () {

    const userInput = document.getElementById("message-input").value;

    if (userInput.trim() === "") return;

    // Display user input in chat box

    const userMessage = document.createElement("div");

    userMessage.textContent = "You: " + userInput;

    document.getElementById("chatbot-messages").appendChild(userMessage);

    // Send the user input to the Flask backend

    try {

        const response = await fetch('http://127.0.0.1:5001//api', {

            method: 'POST',

            headers: { 'Content-Type': 'application/json' },

            body: JSON.stringify({ message: userInput })

        });

        const data = await response.json();

        const botMessage = document.createElement("div");

        botMessage.textContent = "Bot: " + data.response;

        document.getElementById("chatbot-messages").appendChild(botMessage);

    } catch (error) {

        console.error("Error communicating with chatbot:", error);

    }

    // Clear the input field

    document.getElementById("message-input").value = "";

});

**static/Signup.css:**

**Signup.css:**

/\* From Uiverse.io by JohnnyCSilva \*/

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  background-color: white;

}

body{

    background-color: #f0f0f0;

    /\* background-image: url("../static/assets/dashboard.jpeg"); \*/

    /\* border: none;

    background: none; \*/

}

.text-overlay{

  position: absolute;

  top: 10%;

  left: 50%;

  transform: translate(-50%, -50%); /\* Centers the text \*/

  color: white; /\* Text color \*/

  font-size: 3rem; /\* Adjust font size \*/

  font-family: 'Poppins', serif;

  text-align: center;

  background: rgba(0, 0, 0, 0.5); /\* Optional background for better readability \*/

  padding: 10px 20px;

  border-radius: 10px; /\* Optional rounded corners \*/

}

.text-overlay-p{

  position: absolute;

  top: 15%;

  left: 50%;

  transform: translate(-50%, -50%); /\* Centers the text \*/

  color: rgb(197, 197, 197); /\* Text color \*/

  font-size: 1rem; /\* Adjust font size \*/

  font-family: 'Poppins', serif;

  text-align: center;

  background: rgba(0, 0, 0, 0.5); /\* Optional background for better readability \*/

  padding: 10px 20px;

  border-radius: 10px; /\* Optional rounded corners \*/

}

.login-img{

  width: 100%;

  height: 100%;

  object-fit: cover;    /\* Ensures the image covers the container \*/

  display: block;       /\* Removes default spacing from inline images \*/

  border: none;         /\* Removes any default border \*/

}

.main{

  display: flex;

  width: 100%;

  height:625px;

}

.left{

  position: relative;

  width: 58%;

  height: 100vh;

  background-color: #2d79f3;

  margin: 0;

  padding: 0;

}

.form {

    display: flex;

    flex-direction: column;

    margin: 0 auto;

    margin-top: 5%;

    gap: 10px;

    background-color: #ffffff;

    padding: 30px;

    width: 30%;

    border-radius: 20px;

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

  }

  ::placeholder {

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

  }

  .form button {

    align-self: flex-end;

  }

  .flex-column > label {

    color: #151717;

    font-weight: 600;

  }

  .inputForm {

    border: 1.5px solid #ecedec;

    border-radius: 10px;

    height: 50px;

    display: flex;

    align-items: center;

    padding-left: 10px;

    transition: 0.2s ease-in-out;

  }

  .input {

    margin-left: 10px;

    border-radius: 10px;

    border: none;

    width: 100%;

    height: 100%;

  }

  .input:focus {

    outline: none;

  }

  .inputForm:focus-within {

    border: 1.5px solid #2d79f3;

  }

  .flex-row {

    display: flex;

    flex-direction: row;

    align-items: center;

    gap: 10px;

    justify-content: space-between;

  }

  .flex-row > div > label {

    font-size: 14px;

    color: black;

    font-weight: 400;

  }

  .span {

    font-size: 14px;

    margin-left: 5px;

    color: #2d79f3;

    font-weight: 500;

    cursor: pointer;

  }

  .button-submit {

    margin: 20px 0 10px 0;

    background-color: #151717;

    border: none;

    color: white;

    font-size: 15px;

    font-weight: 500;

    border-radius: 10px;

    height: 50px;

    width: 100%;

    cursor: pointer;

  }

  .p {

    text-align: center;

    color: black;

    font-size: 14px;

    margin: 5px 0;

  }

  .btn {

    margin-top: 10px;

    width: 100%;

    height: 50px;

    border-radius: 10px;

    display: flex;

    justify-content: center;

    align-items: center;

    font-weight: 500;

    gap: 10px;

    border: 1px solid #ededef;

    background-color: white;

    cursor: pointer;

    transition: 0.2s ease-in-out;

  }

  .btn:hover {

    border: 1px solid #2d79f3;

    ;

  }

  .name-login{

    text-align: center;

    color: rgb(156, 154, 154);

  }

  .span-name-login{

    color: rgb(30, 175, 124);

  }

**static/styles.css:**

**styles.css:**

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Georama:ital,wght@0,100..900;1,100..900&family=Outfit:wght@100..900&family=Playwrite+MX+Guides&family=Playwrite+VN:wght@100..400&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Roboto:ital,wght@0,100..900;1,100..900&display=swap');

\*{

    box-sizing: border-box;

    margin: 0;

    padding: 0;

    scrollbar-width: none;

}

/\* form \*/

.predict{

    text-align: center;

    margin-top: 50px;

    background-color: #eae9e9;

    padding-top: 20px;

    padding-bottom: 40px;

    margin-left: 200px;

    margin-right: 200px;

    border-radius: 20px;

    height: 400px;

}

input[type="file"] {

    margin: 20px;

    background-color: #d3d3d3;

    width: 700px;

    margin-top: 50px;

}

input[type="submit"] {

    padding: 10px 20px;

    font-size: 16px;

}

header{

    /\* background-color: #31cab3; \*/

   background-image: linear-gradient(270deg, rgb(32, 209, 171), rgb(7, 84, 65));

    color: aliceblue;

    padding: 5px;

    box-shadow: 1px 1px 10px 2px #d9d8d8;

    height: 65px;

    border-radius: 50px 50px 50px 50px;

    padding-top: 10px;

    margin-top: 10px;

    margin-left: 150px;

    margin-right: 150px;

}

nav{

    display: flex;

    justify-content: space-around;

    gap: 30px;

    height: 55px;

    margin-top: 0;

}

/\* nav .head{

    margin-left: 100px;

}

nav .head1{

    margin-left: 750px;

} \*/

.form{

    background-color: #ffffff;

    padding: 1px;

    padding-bottom: 40px;

    border-radius:0px 0px 10px 10px ;

}

.h1-xray{

    color: #2c6b7a;

    margin-top: 40px;

}

.btn{

    background-color: #31cab3;

    border-color: #31cab3;

    border-radius: 10px;

    margin-top: 120px;

    color: white;

    width: 200px;

    padding: 10px;

    font-family: "Poppins", serif;

}

.img{

    background-color: #e8e8e8;

    border-radius: 10px;

    padding: 10px;

}

.head{

    display: flex;

    gap: 50px;

    margin: 0px;

    padding: 0;

}

.heads{

    display: flex;

    background-color:#31cab3;

    background-color: transparent;

    padding: 5px;

    border-radius: 5px;

    color: #e8e8e8;

}

.heads:hover{

    /\* background-color: #2cb5a0; \*/

    transform: rotate(5deg);

    transform: scale(1.1);

    color: #ffffff;

}

.pic{

    /\* display: flex; \*/

    justify-content: center;

    align-items: center;

    text-align: center;

    padding-top: 0px;

    gap: 60px;

    padding-bottom: 50px;

}

.title{

    color: #076b5c;

    font-family: "Poppins", serif;

    font-weight: 300;

    font-style: normal;

    margin-top: 30px;

}

.card-title{

    font-family: "Poppins", serif;

    color: #076b5c;

}

.head1{

    /\* margin-left: 50em; \*/

    background-color: transparent;

    margin-bottom: 0px;

    display: flex;

    padding: 0;

    gap: 50px;

}

.name{

    display: flex;

    /\* background-color:#31cab3; \*/

    background-color: transparent;

    color: #e8e8e8;

    font-size: 25px;

    margin-right: 0px;

    font-family: "Poppins", serif;

  font-weight: 400;

  font-style: normal;

    /\* margin-top: 10px; \*/

}

.cards{

    display: flex;

    gap: 50px;

    justify-content: center;

    margin-right: 30px;

    padding-bottom: 30px;

    margin-top: 50px;

}

.card-text{

    color: #b4b4b4;

}

/\* .cards-1{

    display: flex;

    gap: 50px;

    justify-content: center;

    margin-right: 30px;

} \*/

a{

    text-decoration: none;

    color: black;

}

footer{

    background-color: #ffffff;

    text-align: center;

    justify-content: center;

    align-items: center;

    margin-top: 100px;

}

.icons{

    width: 35px;

}

.main\_img{

    border-radius: 20px 20px 50px 50px;

    width: 1000px;

    /\* border-radius: 86% 79% 89% 51% / 82% 39% 100% 86% ; \*/

}

.span-title{

    color: #44ff00;

    font-style: italic;

}

.span-subtitle{

    color: #b0adad;

    font-weight: 0;

}

.main-title{

    font-size: 50px;

    font-family:Georgia, 'Times New Roman', Times, serif;

  background: -webkit-linear-gradient(45deg,#39a696,rgb(0, 255, 136), #027459,  rgb(5, 5, 5));

  -webkit-background-clip: text;

  -webkit-text-fill-color: transparent;

}

.span-cards{

    color: #31cab3;

}

.span-dent{

    color: #0b7a87;

}

.appoint-logo{

    padding-bottom: 18px;

}

.btn.btn-outline-primary{

    background: transparent;

    border-color: white;

    margin-top:0;

}

.main-logo{

    color: #66c9b2;

    font-weight: 400;

}

.head2{

    color: #cacbcb;

    font-style: italic;

    font-style: "Poppins", serif;

    font-size: 25px;

    font-weight: bold;

}

.messsageDiv{

    background-color: hsl(327,90%,28%);

    color: white;

    padding: 10px 20px;

    margin: 10px;

    border-radius: 5px;

    font-size: 1rem;

    opacity: 0;

    animation: fadeOut 7s forwards;

}

.user\_profile{

    width: 45px;

    height: 45px;

}

**templates/about\_us.html:**

**about\_us.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>About\_us</title>

    <link rel="stylesheet" href="../static/about\_us.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

</head>

<body>

    <div class="top">

        <div class="top-left">

            <h1 class="h1-title">About us</h1>

            <p class="p-text">Feel free to contact us.</p>

        </div>

    </div>

</body>

</html>

**templates/contact\_us.html:**

**contact\_us.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Contact\_us</title>

    <link rel="stylesheet" href="../static/contact\_us.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

</head>

<body>

    <div class="top">

        <div class="top-left">

            <h1 class="h1-title">Contact us</h1>

            <p class="p-text">Feel free to contact us.</p>

        </div>

    </div>

    <div class="card mb-3">

        <div class="row g-0">

          <div class="col-md-4">

            <img src="../static/assets/mypic.jpg" class="img-fluid rounded-start" alt="...">

          </div>

          <div class="col-md-8">

            <div class="card-body">

              <h1 class="card-title"><span class="my-name">Aryan</span> Palaspagar</h1>

              <p class="position"><span>Senior</span> Developer</p>

              <div class="options">

                <ul id="ul-1" class="ul-icons">

                  <li id="phone" class="call-mail">

                    <div class="phone-no">

                      <img class="phone-mail" src="../static/assets/mobile.png">

                      <p class="phone-text">+918080702916</p>

                    </div>

                  </li>

                  <li class="call-mail">

                    <div class="email-id">

                      <img class="phone-mail" src="../static/assets/email.png">

                      <p class="email-text">aryanpalaspagar30@gmail.com</p>

                    </div>

                  </li>

                </ul>

                <ul class="ul-icons">

                    <!-- <a href=""><li class="call-mail"><img src="../static/assets/mobile.png"></li></a>

                    <a href=""><li class="call-mail"><img src="../static/assets/email.png"></li></a> -->

                    <a href="https://github.com/i-am-ap"><li><img src="../static/assets/github.png" target="\_blank"></li></a>

                    <a href="https://www.linkedin.com/in/aryan-palaspagar-9a5985202/" target="\_blank"><li><img src="../static/assets/linkedin.png"></li></a>

                    <a href="https://www.instagram.com/ap\_live\_now/" target="\_blank"><li><img src="../static/assets/insta.png"></li></a>

                    <a href="https://www.facebook.com/aryan.a.palaspagar" target="\_blank"><li><img src="../static/assets/facebook.png"></li></a>

                </ul>

              </div>

              <p class="info">Full Stack Developer with handson experience in Python and Machine Learning.</p>

              <!-- <p class="card-text"><small class="text-body-secondary">Last updated 3 mins ago</small></p> -->

            </div>

          </div>

        </div>

      </div>

</body>

</html>

**templates/dentalimg.html:**

**dentalimg.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Orthodontic Treatment</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/dentalimg.css">

    <link rel="icon" href="{{ url\_for('static', filename='assets/favicon.ico') }}" type="image/x-icon">

</head>

<body>

    <nav class="navbar bg-body-tertiary">

        <div class="container-fluid">

          <a class="navbar-brand" href="#">

            <img src="../static/assets/or.png" alt="Logo" width="30" class="d-inline-block align-text-top">

            <span class="ortho-name">Dental Image</span> Treatment

          </a>

          <button class="btn btn-primary" onclick="goBack()">Back</button>

        </div>

      </nav>

      <div class="main">

        <div class="left">

          <img class="img-left" src="../static/assets/dental.jpg" alt="">

        </div>

        <div class="right">

          <h1 class="title-name"><span class="title-span">Dental Image</span> Diagnosis</h1>

          <p class="or-p">Upload image to predict</p>

          <form class="predict" action="/predict2" method="post" enctype="multipart/form-data">

            <input type="file" name="file" accept="image/\*" class="img" required>

            <br>

            <!-- <input type="submit" value="Upload and Predict" class="btn"> -->

            <button class="btn btn-primary">Predict</button>

          </form>

        </div>

      </div>

      <!-- <footer>

        <p>&copy; 2024 SmartDentAi. All rights reserved.</p>

    </footer> -->

    <!-- Footer -->

<footer class="text-center text-lg-start bg-body-tertiary text-muted">

  <!-- Section: Social media -->

  <section class="d-flex justify-content-center justify-content-lg-between p-4 border-bottom">

    <!-- Left -->

    <div class="me-5 d-none d-lg-block">

      <span class="foot-span">Get connected with us on social networks:</span>

    </div>

    <!-- Left -->

    <!-- Right -->

    <div>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-facebook-f"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-twitter"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-google"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-instagram"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-linkedin"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-github"></i>

      </a>

    </div>

    <!-- Right -->

  </section>

  <!-- Section: Social media -->

  <!-- Section: Links  -->

  <section class="">

    <div class="container text-center text-md-start mt-5">

      <!-- Grid row -->

      <div class="row mt-3">

        <!-- Grid column -->

        <div class="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">

          <!-- Content -->

          <h6 class="text-uppercase fw-bold mb-4">

            <i class="fas fa-gem me-3"></i>Company name

          </h6>

          <p>

            Here you can use rows and columns to organize your footer content. Lorem ipsum

            dolor sit amet, consectetur adipisicing elit.

          </p>

        </div>

        <!-- Grid column -->

        <!-- Grid column -->

        <div class="col-md-2 col-lg-2 col-xl-2 mx-auto mb-4">

          <!-- Links -->

          <h6 class="text-uppercase fw-bold mb-4">

            Products

          </h6>

          <p>

            <a href="#!" class="text-reset">Angular</a>

          </p>

          <p>

            <a href="#!" class="text-reset">React</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Vue</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Laravel</a>

          </p>

        </div>

        <!-- Grid column -->

        <!-- Grid column -->

        <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">

          <!-- Links -->

          <h6 class="text-uppercase fw-bold mb-4">

            Useful links

          </h6>

          <p>

            <a href="#!" class="text-reset">Pricing</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Settings</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Orders</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Help</a>

          </p>

        </div>

        <!-- Grid column -->

        <!-- Grid column -->

        <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">

          <!-- Links -->

          <h6 class="text-uppercase fw-bold mb-4">Contact</h6>

          <p><i class="fas fa-home me-3"></i> New York, NY 10012, US</p>

          <p>

            <i class="fas fa-envelope me-3"></i>

            info@example.com

          </p>

          <p><i class="fas fa-phone me-3"></i> + 01 234 567 88</p>

          <p><i class="fas fa-print me-3"></i> + 01 234 567 89</p>

        </div>

        <!-- Grid column -->

      </div>

      <!-- Grid row -->

    </div>

  </section>

  <!-- Section: Links  -->

  <!-- Copyright -->

  <div class="text-center p-4" style="background-color: rgba(0, 0, 0, 0.05);">

    © 2021 Copyright:

    <a class="text-reset fw-bold" href="https://mdbootstrap.com/">MDBootstrap.com</a>

  </div>

  <!-- Copyright -->

</footer>

<!-- Footer -->

      <script>

        function goBack() {

          window.history.back();

      }

      </script>

</body>

</html>

**templates/dentalxray.html:**

**dentalxray.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/styles.css">

    <!-- <link rel="stylesheet" href="{{ url\_for('static', filename='styles.css') }}"> -->

    <link rel="stylesheet" href="../static//chatbot.css">

    <!-- <link rel="stylesheet" href="{{ url\_for('static', filename='chatbot.css') }}"> -->

    <script src="../static/scripts.js"></script>

    <title>Document</title>

</head>

<body>

    <div class="form">

        <form action="/predict" method="post" enctype="multipart/form-data">

            <h1 style="color: #2c6b7a;">Xray Prediction</h1>

            <p>Upload your file here!</p>

        <input type="file" name="file" accept="image/\*" class="img" required>

        <br>

        <input type="submit" value="Upload and Predict" class="btn">

    </form>

    </div>

</body>

</html>

**templates/home.html:**

**home.html:**

# 📁 templates/home.html -----

<html>

  <head>

    <meta charset="utf-8" />

    <title>Auth0 Example</title>

  </head>

  <body>

    {% if session %}

        <h1>Welcome {{session.userinfo.name}}!</h1>

        <p><a href="/logout">Logout</a></p>

        <div><pre>{{pretty}}</pre></div>

    {% else %}

        <h1>Welcome Guest</h1>

        <p><a href="/login">Login</a></p>

    {% endif %}

  </body>

</html>

**templates/index.html:**

**index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="icon" href="{{ url\_for('static', filename='assets/favicon.ico') }}" type="image/x-icon">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/styles.css">

    <!-- <link rel="stylesheet" href="{{ url\_for('static', filename='styles.css') }}"> -->

    <link rel="stylesheet" href="../static//chatbot.css">

    <script type="module" src="../static/index.js" defer></script>

    <!-- <link rel="stylesheet" href="{{ url\_for('static', filename='chatbot.css') }}"> -->

    <!-- <script src="../static/scripts.js"></script> -->

    <title>X-ray Image Prediction</title>

</head>

<body>

    <header class="header">

        <!-- <h1>Upload X-ray Image for Disease Prediction</h1> -->

        <nav>

            <ul class="head2">

              <a><li class="heads">Smart<span class="main-logo">Dent</span>Ai</li></a>

            </ul>

            <ul class="head">

                <!-- <a href="" class="" style="text-decoration: none;"><li class="heads">Home</li></a> -->

                <a href="" class="" style="text-decoration: none;"><li class="heads"><img class="icons" src="../static/assets/homee.png" alt="Image"></li></a>

                <a href="{{ url\_for('contact') }}" class="" style="text-decoration: none;"><li class="heads"><img class="icons" src="../static/assets/calll.png" alt="Image"></li></a>

                <a href="{{ url\_for('about') }}" class="" style="text-decoration: none;"><li class="heads"><img class="icons" src="../static/assets/about\_uss.png" alt="Image"></li></a>

                <!-- <a href="" class="" style="text-decoration: none;"><li class="heads">Login</li></a>

                <a href="" class="" style="text-decoration: none;"><li class="heads">Logout</li></a> -->

            </ul>

            <ul class="head1">

                <!-- <a href="" style="text-decoration: none;"><li class="name">Smart<span class="span-dent">Dent</span>Ai</li></a> -->

                <a href="{{ url\_for('profile') }}"><img class="user\_profile" src="../static//assets//man.png" alt=""></a>

                <button class="btn btn-outline-primary" id="logout">Sign out</button>

            </ul>

        </nav>

    </header>

    <div class="pic">

        <div class="title">

            <h1 class="main-title">Dental <span class="span-title">Diagnosis</span> and Treatment Solutions</h1>

            <p style="color: #31cab3;">Empowering Your Smile <span class="span-subtitle">Predict and Prevent Dental Issues with Ease</span></p>

        </div>

        <div class="dent">

            <img class="main\_img" src="../static/assets/sheehan-summit.jpg" alt="Image">

        </div>

    </div>

    <!-- Form to upload the X-ray image -->

    <div class="form">

        <form class="predict" action="/predict" method="post" enctype="multipart/form-data">

            <h1 class="h1-xray">Disease Detection and Diagnosis of Xray</h1>

            <p>Upload your file here!</p>

        <input type="file" name="file" accept="image/\*" class="img" required>

        <br>

        <!-- <input type="submit" value="Upload and Predict" class="btn"> -->

        <button class="btn btn-primary">Predict</button>

    </form>

    </div>

    <!-- Form to upload the Dental image -->

    <!-- <div class="services"> -->

        <div class="cards">

            <!-- <div class="card text-center mb-3" style="width: 18rem;">

                <div class="card-body">

                  <h5 class="card-title">Oral Treatment</h5>

                  <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>

                  <a href="oraldetection.html" class="btn btn-primary">Click to Start</a>

                </div>

              </div> -->

              <div class="card text-center mb-3" style="width: 18rem;">

                <div class="card-body">

                  <h5 class="card-title"><span class="span-cards">Appointment</span> Booking</h5>

                  <img class="appoint-logo" src="../static/assets/appointt.png" alt="" width="150px">

                  <p class="card-text">Choose your preferred dentist, book an appointment with ease, and get timely reminders—all in just a few clicks</p>

                  <!-- <a href="appointment.html" class="btn btn-primary">Click to Start</a> -->

                  <form action="/appointment" method="post">

                    <button class="btn btn-primary">Click to Start</button>

                </form>

                </div>

              </div>

              <div class="card text-center mb-3" style="width: 18rem;">

                <div class="card-body">

                  <h5 class="card-title"><span class="span-cards">Orthodontics</span> Treatment</h5>

                  <img src="../static/assets/orthod.png" alt="" width="150px">

                  <p class="card-text">Upload an image of your misaligned teeth and preview your perfect smile with AI-powered alignment results.</p>

                  <!-- <a href="orthodontic.html" class="btn btn-primary" action="/predict" method="post">Click to Start</a> -->

                <form action="/ortho\_predict" method="post">

                    <button class="btn btn-primary">Click to Start</button>

                </form>

                </div>

              </div>

              <div class="card text-center mb-3" style="width: 18rem;">

                <div class="card-body">

                  <h5 class="card-title"><span class="span-cards">Dental</span> Diagnosis</h5>

                  <img src="../static/assets/teeth.png" alt="" width="150px">

                  <p class="card-text">Upload your dental image and let AI analyze it for potential issues, providing quick and accurate treatment suggestions.</p>

                  <!-- <a href="orthodontic.html" class="btn btn-primary" action="/predict" method="post">Click to Start</a> -->

                <form action="/dentalimg" method="post">

                    <button class="btn btn-primary">Click to Start</button>

                </form>

                </div>

              </div>

        <!-- </div> -->

        <!-- <div class="cards-1"> -->

            <div class="card text-center mb-3" style="width: 18rem;">

                <div class="card-body">

                  <h5 class="card-title"><span class="span-cards">Remote</span> Consultation</h5>

                  <img src="../static/assets/remote.png" alt="" width="150px">

                  <p class="card-text">  Connect with a dentist via video call, get personalized guidance, and discuss your concerns from the comfort of your home.</p>

                  <!-- <a href="remoteconsult.html" class="btn btn-primary">Click to Start</a> -->

                  <form action="/remote" method="post">

                    <button class="btn btn-primary">Click to Start</button>

                </form>

                </div>

              </div>

              <!-- <div class="card text-center mb-3" style="width: 18rem;">

                <div class="card-body">

                  <h5 class="card-title">Dental Xray Analysis</h5>

                  <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>

                  <a href="dentalxray.html" class="btn btn-primary">Go somewhere</a>

                </div>

              </div> -->

        </div>

    <!-- </div> -->

    <div>

        <!-- Chatbot Button -->

        <div class="chatbot-icon" onclick="toggleChatbot()">

            <img src="../static/assets/chatbot-icon.png" alt="Chat Icon" width="150">

        </div>

        <!-- Chatbot Window -->

        <div class="chatbot-window" id="chatbot-window" style="display: none;">

            <div class="chatbot-header">

                <h3 class="bot-msg">Chat with us!</h3>

                <button class="msg-close-btn" onclick="toggleChatbot()">X</button>

            </div>

            <div class="chat-box" id="chatbot-messages">

                <!-- Chat messages will be added here dynamically -->

            </div>

            <div class="form-group mt-3">

                <textarea class="form-control" rows="3" placeholder="Type your message here" id="message-input"></textarea>

            </div>

            <button type="button" class="btn btn-primary" id="send-btn">Send</button>

        </div>

    </div>

    <!-- <footer>

        <p>&copy; 2024 SmartDentAi. All rights reserved.</p>

    </footer> -->

    <!-- Footer -->

<footer class="text-center text-lg-start bg-body-tertiary text-muted">

    <!-- Section: Social media -->

    <section class="d-flex justify-content-center justify-content-lg-between p-4 border-bottom">

      <!-- Left -->

      <div class="me-5 d-none d-lg-block">

        <span>Get connected with us on social networks:</span>

      </div>

      <!-- Left -->

      <!-- Right -->

      <div>

        <a href="" class="me-4 text-reset">

          <i class="fab fa-facebook-f"></i>

        </a>

        <a href="" class="me-4 text-reset">

          <i class="fab fa-twitter"></i>

        </a>

        <a href="" class="me-4 text-reset">

          <i class="fab fa-google"></i>

        </a>

        <a href="" class="me-4 text-reset">

          <i class="fab fa-instagram"></i>

        </a>

        <a href="" class="me-4 text-reset">

          <i class="fab fa-linkedin"></i>

        </a>

        <a href="" class="me-4 text-reset">

          <i class="fab fa-github"></i>

        </a>

      </div>

      <!-- Right -->

    </section>

    <!-- Section: Social media -->

    <!-- Section: Links  -->

    <section class="">

      <div class="container text-center text-md-start mt-5">

        <!-- Grid row -->

        <div class="row mt-3">

          <!-- Grid column -->

          <div class="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">

            <!-- Content -->

            <h6 class="text-uppercase fw-bold mb-4">

              <i class="fas fa-gem me-3"></i>Smartdentai

            </h6>

            <p>

              Here you can use rows and columns to organize your footer content. Lorem ipsum

              dolor sit amet, consectetur adipisicing elit.

            </p>

          </div>

          <!-- Grid column -->

          <!-- Grid column -->

          <div class="col-md-2 col-lg-2 col-xl-2 mx-auto mb-4">

            <!-- Links -->

            <h6 class="text-uppercase fw-bold mb-4">

              Products

            </h6>

            <p>

              <a href="#!" class="text-reset">Javascript</a>

            </p>

            <p>

              <a href="#!" class="text-reset">Python</a>

            </p>

            <p>

              <a href="#!" class="text-reset">Bootstrap</a>

           </p>

            <p>

              <a href="#!" class="text-reset">SQL</a>

            </p>

          </div>

          <!-- Grid column -->

          <!-- Grid column -->

          <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">

            <!-- Links -->

            <h6 class="text-uppercase fw-bold mb-4">

              Useful links

            </h6>

            <p>

              <a href="#!" class="text-reset">Sign in</a>

            </p>

            <p>

              <a href="#!" class="text-reset">Sign up</a>

            </p>

            <p>

              <a href="#!" class="text-reset">About</a>

            </p>

            <p>

              <a href="#!" class="text-reset">Help</a>

            </p>

          </div>

          <!-- Grid column -->

          <!-- Grid column -->

          <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">

            <!-- Links -->

            <h6 class="text-uppercase fw-bold mb-4">Contact</h6>

            <p><i class="fas fa-home me-3"></i> Mumbai, India</p>

            <p>

              <i class="fas fa-envelope me-3"></i>

              smartdentai.com

            </p>

            <p><i class="fas fa-phone me-3"></i> + 01 234 567 88</p>

            <p><i class="fas fa-print me-3"></i> + 01 234 567 89</p>

          </div>

          <!-- Grid column -->

        </div>

        <!-- Grid row -->

      </div>

    </section>

    <!-- Section: Links  -->

    <!-- Copyright -->

    <div class="text-center p-4" style="background-color: rgba(0, 0, 0, 0.05);">

      © 2024 SmartDentAi. All rights reserved.

      <!-- <a class="text-reset fw-bold" href="https://mdbootstrap.com/">MDBootstrap.com</a>

    </div> -->

    <!-- Copyright -->

  </footer>

  <!-- Footer -->

    <!-- <script src="scripts.js"></script> -->

    <script src="{{ url\_for('static', filename='scripts.js') }}"></script>

</body>

</html>

**templates/login.html:**

**login.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="../static/login.css">

    <title>Document</title>

    <script type="module" src="../static/login.js" defer></script>

</head>

<body>

    <!-- From Uiverse.io by JohnnyCSilva -->

    <div class="main">

      <div class="left">

        <img class="login-img" src="../static/assets/login-bg.png" alt="Background Image" >

        <div class="text-overlay">Welcome</div>

        <p class="text-overlay-p">Hi, Login to start with SmartDentAi.</p>

      </div>

      <form class="form" action="/signup" method="post" enctype="multipart/form-data">

        <div id="signIpMessage" class="messsageDiv" style="display:none;"></div>

          <h2 class="name-login">Smart<span class="span-name-login">Dent</span>Ai</h2>

        <div class="flex-column">

          <label>Email </label></div>

        <div class="inputForm">

          <svg xmlns="http://www.w3.org/2000/svg" width="20" viewBox="0 0 32 32" height="20"><g data-name="Layer 3" id="Layer\_3"><path d="m30.853 13.87a15 15 0 0 0 -29.729 4.082 15.1 15.1 0 0 0 12.876 12.918 15.6 15.6 0 0 0 2.016.13 14.85 14.85 0 0 0 7.715-2.145 1 1 0 1 0 -1.031-1.711 13.007 13.007 0 1 1 5.458-6.529 2.149 2.149 0 0 1 -4.158-.759v-10.856a1 1 0 0 0 -2 0v1.726a8 8 0 1 0 .2 10.325 4.135 4.135 0 0 0 7.83.274 15.2 15.2 0 0 0 .823-7.455zm-14.853 8.13a6 6 0 1 1 6-6 6.006 6.006 0 0 1 -6 6z"></path></g></svg>

          <input placeholder="Enter your Email" class="input" type="text" id="email" required>

         </div>

    <div class="flex-column">

      <label>Password </label></div>

      <div class="inputForm">

        <svg xmlns="http://www.w3.org/2000/svg" width="20" viewBox="-64 0 512 512" height="20"><path d="m336 512h-288c-26.453125 0-48-21.523438-48-48v-224c0-26.476562 21.546875-48 48-48h288c26.453125 0 48 21.523438 48 48v224c0 26.476562-21.546875 48-48 48zm-288-288c-8.8125 0-16 7.167969-16 16v224c0 8.832031 7.1875 16 16 16h288c8.8125 0 16-7.167969 16-16v-224c0-8.832031-7.1875-16-16-16zm0 0"></path><path d="m304 224c-8.832031 0-16-7.167969-16-16v-80c0-52.929688-43.070312-96-96-96s-96 43.070312-96 96v80c0 8.832031-7.167969 16-16 16s-16-7.167969-16-16v-80c0-70.59375 57.40625-128 128-128s128 57.40625 128 128v80c0 8.832031-7.167969 16-16 16zm0 0"></path></svg>

        <input placeholder="Enter your Password" class="input" type="password" id="password" required>

      </div>

    <div class="flex-row">

      <div>

      <input type="radio">

      <label>Remember me </label>

      </div>

      <!-- <span class="span">Forgot password?</span> -->

       <a href="" id="reset"><span class="span">Forgot password?</span></a>

    </div>

    <button class="button-submit" id="submit">Sign In</button>

    <p class="p">Don't have an account? <span class="span">

      <a href="{{ url\_for('start') }}">Sign up</a></span></p>

    </span>

    </p><p class="p line">Or With</p>

    <div class="flex-row">

      <button class="btn google">

        <svg xml:space="preserve" style="enable-background:new 0 0 512 512;" viewBox="0 0 512 512" y="0px" x="0px" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.w3.org/2000/svg" id="Layer\_1" width="20" version="1.1">

<path d="M113.47,309.408L95.648,375.94l-65.139,1.378C11.042,341.211,0,299.9,0,256

  c0-42.451,10.324-82.483,28.624-117.732h0.014l57.992,10.632l25.404,57.644c-5.317,15.501-8.215,32.141-8.215,49.456

  C103.821,274.792,107.225,292.797,113.47,309.408z" style="fill:#FBBB00;"></path>

<path d="M507.527,208.176C510.467,223.662,512,239.655,512,256c0,18.328-1.927,36.206-5.598,53.451

  c-12.462,58.683-45.025,109.925-90.134,146.187l-0.014-0.014l-73.044-3.727l-10.338-64.535

  c29.932-17.554,53.324-45.025,65.646-77.911h-136.89V208.176h138.887L507.527,208.176L507.527,208.176z" style="fill:#518EF8;"></path>

<path d="M416.253,455.624l0.014,0.014C372.396,490.901,316.666,512,256,512

  c-97.491,0-182.252-54.491-225.491-134.681l82.961-67.91c21.619,57.698,77.278,98.771,142.53,98.771

  c28.047,0,54.323-7.582,76.87-20.818L416.253,455.624z" style="fill:#28B446;"></path>

<path d="M419.404,58.936l-82.933,67.896c-23.335-14.586-50.919-23.012-80.471-23.012

  c-66.729,0-123.429,42.957-143.965,102.724l-83.397-68.276h-0.014C71.23,56.123,157.06,0,256,0

  C318.115,0,375.068,22.126,419.404,58.936z" style="fill:#F14336;"></path>

</svg>

        Google

      </button><button class="btn apple">

<svg xml:space="preserve" style="enable-background:new 0 0 22.773 22.773;" viewBox="0 0 22.773 22.773" y="0px" x="0px" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.w3.org/2000/svg" id="Capa\_1" width="20" height="20" version="1.1"> <g> <g> <path d="M15.769,0c0.053,0,0.106,0,0.162,0c0.13,1.606-0.483,2.806-1.228,3.675c-0.731,0.863-1.732,1.7-3.351,1.573 c-0.108-1.583,0.506-2.694,1.25-3.561C13.292,0.879,14.557,0.16,15.769,0z"></path> <path d="M20.67,16.716c0,0.016,0,0.03,0,0.045c-0.455,1.378-1.104,2.559-1.896,3.655c-0.723,0.995-1.609,2.334-3.191,2.334 c-1.367,0-2.275-0.879-3.676-0.903c-1.482-0.024-2.297,0.735-3.652,0.926c-0.155,0-0.31,0-0.462,0 c-0.995-0.144-1.798-0.932-2.383-1.642c-1.725-2.098-3.058-4.808-3.306-8.276c0-0.34,0-0.679,0-1.019 c0.105-2.482,1.311-4.5,2.914-5.478c0.846-0.52,2.009-0.963,3.304-0.765c0.555,0.086,1.122,0.276,1.619,0.464 c0.471,0.181,1.06,0.502,1.618,0.485c0.378-0.011,0.754-0.208,1.135-0.347c1.116-0.403,2.21-0.865,3.652-0.648 c1.733,0.262,2.963,1.032,3.723,2.22c-1.466,0.933-2.625,2.339-2.427,4.74C17.818,14.688,19.086,15.964,20.67,16.716z"></path> </g></g></svg>

        Apple

</button></div></form>

    </div>

</body>

</html>

**templates/Main.html:**

**Main.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/main.css">

    <title>Document</title>

</head>

<body>

    <ul class="nav justify-content-center">

        <li class="nav-item">

          <a class="nav-link active" aria-current="page" href="#">About</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="{{ url\_for('contact') }}">Contact us</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="#">Help</a>

        </li>

      </ul>

      <h1 class="name-title">S m a r t <span class="span-name-title"> D e n t </span> a i</h1>

      <form class="Getstart" action="/start" method="post" enctype="multipart/form-data">

         <div class="d-grid gap-2 col-6 mx-auto">

           <button class="btn btn-primary" type="submit">Get Started</button>

        </div>

      </form>

</body>

</html>

**templates/oraldetection.html:**

**oraldetection.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Oral Detection</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/styles.css">

</head>

<body>

    <header class="header">

        <!-- <h1>Upload X-ray Image for Disease Prediction</h1> -->

        <nav>

            <ul class="head">

                <a href="index.html" style="text-decoration: none;"><li class="heads">Back</li></a>

            </ul>

        </nav>

        <nav class="head1">

            <ul>

                <a href="" style="text-decoration: none;"><li class="name">SmartDentAi</li>

            </ul>

        </nav>

    </header>

    <div class="form">

        <!-- Form to upload the X-ray image -->

        <form action="/predict" method="post" enctype="multipart/form-data">

            <h1 style="color: #2c6b7a;">Oral Detection</h1>

            <p>Upload your file here!</p>

        <input type="file" name="file" accept="image/\*" class="img" required>

        <br>

        <input type="submit" value="Upload and Predict" class="btn">

    </form>

    </div>

</body>

</html>

**templates/orthodontic.html:**

**orthodontic.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Orthodontic Treatment</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/Orthodontic.css">

    <link rel="icon" href="{{ url\_for('static', filename='assets/favicon.ico') }}" type="image/x-icon">

</head>

<body>

    <nav class="navbar bg-body-tertiary">

        <div class="container-fluid">

          <a class="navbar-brand" href="#">

            <img src="../static/assets/or.png" alt="Logo" width="30" class="d-inline-block align-text-top">

            <span class="ortho-name">Orthodontic</span> Treatment

          </a>

          <button class="btn btn-primary" onclick="goBack()">Back</button>

        </div>

      </nav>

      <div class="main">

        <div class="left">

          <img class="img-left" src="../static/assets/dr.png" alt="">

        </div>

        <div class="right">

          <h1 class="title-name"><span class="title-span">Orthodontic</span> Treatment</h1>

          <p class="or-p">Upload image to predict</p>

          <form class="predict" action="/ortho\_predict" method="post" enctype="multipart/form-data">

            <input type="file" name="file" accept="image/\*" class="img" required>

            <br>

            <!-- <input type="submit" value="Upload and Predict" class="btn"> -->

            <button class="btn btn-primary">Orthodontic</button>

          </form>

        </div>

      </div>

      <!-- <footer>

        <p>&copy; 2024 SmartDentAi. All rights reserved.</p>

    </footer> -->

    <!-- Footer -->

<footer class="text-center text-lg-start bg-body-tertiary text-muted">

  <!-- Section: Social media -->

  <section class="d-flex justify-content-center justify-content-lg-between p-4 border-bottom">

    <!-- Left -->

    <div class="me-5 d-none d-lg-block">

      <span class="foot-span">Get connected with us on social networks:</span>

    </div>

    <!-- Left -->

    <!-- Right -->

    <div>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-facebook-f"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-twitter"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-google"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-instagram"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-linkedin"></i>

      </a>

      <a href="" class="me-4 text-reset">

        <i class="fab fa-github"></i>

      </a>

    </div>

    <!-- Right -->

  </section>

  <!-- Section: Social media -->

  <!-- Section: Links  -->

  <section class="">

    <div class="container text-center text-md-start mt-5">

      <!-- Grid row -->

      <div class="row mt-3">

        <!-- Grid column -->

        <div class="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">

          <!-- Content -->

          <h6 class="text-uppercase fw-bold mb-4">

            <i class="fas fa-gem me-3"></i>Company name

          </h6>

          <p>

            Here you can use rows and columns to organize your footer content. Lorem ipsum

            dolor sit amet, consectetur adipisicing elit.

          </p>

        </div>

        <!-- Grid column -->

        <!-- Grid column -->

        <div class="col-md-2 col-lg-2 col-xl-2 mx-auto mb-4">

          <!-- Links -->

          <h6 class="text-uppercase fw-bold mb-4">

            Products

          </h6>

          <p>

            <a href="#!" class="text-reset">Angular</a>

          </p>

          <p>

            <a href="#!" class="text-reset">React</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Vue</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Laravel</a>

          </p>

        </div>

        <!-- Grid column -->

        <!-- Grid column -->

        <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">

          <!-- Links -->

          <h6 class="text-uppercase fw-bold mb-4">

            Useful links

          </h6>

          <p>

            <a href="#!" class="text-reset">Pricing</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Settings</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Orders</a>

          </p>

          <p>

            <a href="#!" class="text-reset">Help</a>

          </p>

        </div>

        <!-- Grid column -->

        <!-- Grid column -->

        <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">

          <!-- Links -->

          <h6 class="text-uppercase fw-bold mb-4">Contact</h6>

          <p><i class="fas fa-home me-3"></i> New York, NY 10012, US</p>

          <p>

            <i class="fas fa-envelope me-3"></i>

            info@example.com

          </p>

          <p><i class="fas fa-phone me-3"></i> + 01 234 567 88</p>

          <p><i class="fas fa-print me-3"></i> + 01 234 567 89</p>

        </div>

        <!-- Grid column -->

      </div>

      <!-- Grid row -->

    </div>

  </section>

  <!-- Section: Links  -->

  <!-- Copyright -->

  <div class="text-center p-4" style="background-color: rgba(0, 0, 0, 0.05);">

    © 2021 Copyright:

    <a class="text-reset fw-bold" href="https://mdbootstrap.com/">MDBootstrap.com</a>

  </div>

  <!-- Copyright -->

</footer>

<!-- Footer -->

      <script>

        function goBack() {

          window.history.back();

      }

      </script>

</body>

</html>

**templates/predict.html:**

**predict.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="icon" href="{{ url\_for('static', filename='assets/favicon.ico') }}" type="image/x-icon">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="../static/predict.css">

    <link rel="stylesheet" href="../static/styles.css">

    <title>Prediction and Diagnosis</title>

</head>

<body>

    <header class="header">

        <!-- <h1>Upload X-ray Image for Disease Prediction</h1> -->

        <nav class="nav-predict">

            <ul class="head-predict">

                <a href="{{url\_for('home')}}" style="text-decoration: none;"><li class="heads"><img class="icons-predict" src="../static/assets/homee.png" alt="Image"></li></a>

                <a href="{{url\_for('contact')}}" style="text-decoration: none;"><li class="heads"><img class="icons-predict" src="../static/assets/calll.png" alt="Image"></li></a>

                <!-- <a href="" style="text-decoration: none;"><li class="heads">Signup</li></a>

                <a href="" style="text-decoration: none;"><li class="heads">Login</li></a>

                <a href="" style="text-decoration: none;"><li class="heads">Logout</li></a> -->

            </ul>

            <ul class="head2-predict">

                <a href="" style="text-decoration: none;"><li class="name">SmartDentAi</li>

            </ul>

        </nav>

    </header>

    <h1 class="h1-predict">Predicted Disease: <span class="span-h1-predict">{{ prediction }}</span></h1>

    <!-- Display the uploaded image -->

    <img  class="upload-img" src="{{ url\_for('static', filename=img\_path) }}" alt="Uploaded X-ray Image">

     <!-- <img src="../static/assets/uploads/{{ image\_filename }}" alt="Uploaded X-ray Image"> -->

    <h2 class="h2-predict">Diagnosis Suggestions</h2>

    <!-- Table for detailed diagnosis information -->

    <table>

        <tr class="table-title">

            <th>Category</th>

            <th>Details</th>

        </tr>

        <tr>

            <td class="row-title">Symptoms</td>

            <td>{{ details['symptoms'] }}</td>

        </tr>

        <tr>

            <td class="row-title">Causes</td>

            <td>{{ details['causes'] }}</td>

        </tr>

        <tr>

            <td class="row-title">Treatment</td>

            <td>{{ details['treatment'] }}</td>

        </tr>

        <tr>

            <td class="row-title">Precautions</td>

            <td>{{ details['precautions'] }}</td>

        </tr>

    </table>

    <!-- Footer -->

    <footer class="text-center text-lg-start bg-body-tertiary text-muted">

        <!-- Section: Social media -->

        <section class="d-flex justify-content-center justify-content-lg-between p-4 border-bottom">

          <!-- Left -->

          <div class="me-5 d-none d-lg-block">

            <span>Get connected with us on social networks:</span>

          </div>

          <!-- Left -->

          <!-- Right -->

          <div>

            <a href="" class="me-4 text-reset">

              <i class="fab fa-facebook-f"></i>

            </a>

            <a href="" class="me-4 text-reset">

              <i class="fab fa-twitter"></i>

            </a>

            <a href="" class="me-4 text-reset">

              <i class="fab fa-google"></i>

            </a>

            <a href="" class="me-4 text-reset">

              <i class="fab fa-instagram"></i>

            </a>

            <a href="" class="me-4 text-reset">

              <i class="fab fa-linkedin"></i>

            </a>

            <a href="" class="me-4 text-reset">

              <i class="fab fa-github"></i>

            </a>

          </div>

          <!-- Right -->

        </section>

        <!-- Section: Social media -->

        <!-- Section: Links  -->

        <section class="">

          <div class="container text-center text-md-start mt-5">

            <!-- Grid row -->

            <div class="row mt-3">

              <!-- Grid column -->

              <div class="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">

                <!-- Content -->

                <h6 class="text-uppercase fw-bold mb-4">

                  <i class="fas fa-gem me-3"></i>Smartdentai

                </h6>

                <p>

                  Here you can use rows and columns to organize your footer content. Lorem ipsum

                  dolor sit amet, consectetur adipisicing elit.

                </p>

              </div>

              <!-- Grid column -->

              <!-- Grid column -->

              <div class="col-md-2 col-lg-2 col-xl-2 mx-auto mb-4">

                <!-- Links -->

                <h6 class="text-uppercase fw-bold mb-4">

                  Products

                </h6>

                <p>

                  <a href="#!" class="text-reset">Javascript</a>

                </p>

                <p>

                  <a href="#!" class="text-reset">Python</a>

                </p>

                <p>

                  <a href="#!" class="text-reset">Bootstrap</a>

               </p>

                <p>

                  <a href="#!" class="text-reset">SQL</a>

                </p>

              </div>

              <!-- Grid column -->

              <!-- Grid column -->

              <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">

                <!-- Links -->

                <h6 class="text-uppercase fw-bold mb-4">

                  Useful links

                </h6>

                <p>

                  <a href="#!" class="text-reset">Sign in</a>

                </p>

                <p>

                  <a href="#!" class="text-reset">Sign up</a>

                </p>

                <p>

                  <a href="#!" class="text-reset">About</a>

                </p>

                <p>

                  <a href="#!" class="text-reset">Help</a>

                </p>

              </div>

              <!-- Grid column -->

              <!-- Grid column -->

              <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">

                <!-- Links -->

                <h6 class="text-uppercase fw-bold mb-4">Contact</h6>

                <p><i class="fas fa-home me-3"></i> Mumbai, India</p>

                <p>

                  <i class="fas fa-envelope me-3"></i>

                  smartdentai.com

                </p>

                <p><i class="fas fa-phone me-3"></i> + 01 234 567 88</p>

                <p><i class="fas fa-print me-3"></i> + 01 234 567 89</p>

              </div>

              <!-- Grid column -->

            </div>

            <!-- Grid row -->

          </div>

        </section>

        <!-- Section: Links  -->

        <!-- Copyright -->

        <div class="text-center p-4" style="background-color: rgba(0, 0, 0, 0.05);">

          © 2024 SmartDentAi. All rights reserved.

          <!-- <a class="text-reset fw-bold" href="https://mdbootstrap.com/">MDBootstrap.com</a>

        </div> -->

        <!-- Copyright -->

      </footer>

      <!-- Footer -->

</body>

</html>

**templates/Profile.html:**

**Profile.html:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Profile</title>

    <link

      rel="icon"

      href="{{ url\_for('static', filename='assets/favicon.ico') }}"

      type="image/x-icon"

    />

    <link

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"

      rel="stylesheet"

      integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"

      crossorigin="anonymous"

    />

    <script

      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

      integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"

      crossorigin="anonymous"

    ></script>

    <link rel="stylesheet" href="../static/styles.css" />

    <link rel="stylesheet" href="../static/chatbot.css" />

    <script type="module" src="../static/profile.js" defer></script>

  </head>

  <body>

    <header class="header">

      <nav>

        <ul class="head2">

          <a>

            <li class="heads">

              Smart<span class="main-logo">Dent</span>Ai

            </li>

          </a>

        </ul>

        <ul class="head">

          <a href="{{ url\_for('home') }}" class="" style="text-decoration: none">

            <li class="heads">

              <img class="icons" src="../static/assets/homee.png" alt="Image" />

            </li>

          </a>

          <a href="{{ url\_for('contact') }}" class="" style="text-decoration: none">

            <li class="heads">

              <img class="icons" src="../static/assets/calll.png" alt="Image" />

            </li>

          </a>

          <a href="{{ url\_for('about') }}" class="" style="text-decoration: none">

            <li class="heads">

              <img class="icons" src="../static/assets/about\_uss.png" alt="Image" />

            </li>

          </a>

        </ul>

        <ul class="head1">

          <button class="btn btn-outline-primary" id="logout">Sign out</button>

        </ul>

      </nav>

    </header>

    <div class="container py-5">

      <div class="card mb-3">

        <div class="card-body">

          <h4 class="card-title">User Profile</h4>

          <div class="userprofile">

            <p class="strong">

              <strong>Name:</strong> <span id="username"></span>

            </p>

            <p>

              <strong>Email:</strong> <span id="email"></span>

            </p>

            <p>

              <strong>Profile Picture:</strong>

            </p>

            <img

              src="{{ url\_for('static', filename='assets/default-profile.png') }}"

              alt="Profile Picture"

              id="userProfilePicture"

              class="img-fluid rounded-circle mb-3"

              style="width: 100px; height: 100px;"

            />

            <!-- Added file input and upload button -->

            <div class="mt-3">

              <input type="file" id="profilePictureInput" accept="image/\*" class="form-control mb-2" />

              <button id="uploadProfilePicture" class="btn btn-primary">Upload Profile Picture</button>

            </div>

          </div>

        </div>

      </div>

    </div>

    <!-- User History Section -->

    <div class="card">

      <div class="card-body">

          <h4 class="card-title">User History</h4>

          <table class="table table-striped">

              <thead>

                  <tr>

                      <th>Sr. No.</th>

                      <th>Input Image</th>

                      <th>Output Result</th>

                  </tr>

              </thead>

              <tbody id="userHistory">

                  <!-- Example Row -->

                  <tr>

                      <td>1</td>

                      <td>

                          <img src="path/to/input-image.jpg" alt="Input Preview" class="img-thumbnail" style="width: 100px; height: 100px;" />

                          <br>

                          <a href="path/to/input-image.jpg" download class="btn btn-sm btn-outline-primary mt-2">Download</a>

                      </td>

                      <td>

                          <img src="path/to/output-image.jpg" alt="Output Preview" class="img-thumbnail" style="width: 100px; height: 100px;" onerror="this.style.display='none';" />

                          <span id="outputText">Output Text Example</span>

                          <br>

                          <a href="path/to/output-image.jpg" download class="btn btn-sm btn-outline-primary mt-2">Download</a>

                      </td>

                  </tr>

              </tbody>

          </table>

      </div>

  </div>

  </body>

</html>

**templates/Signup.html:**

**Signup.html:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link

      rel="stylesheet"

      href="{{ url\_for('static', filename='Signup.css') }}"

    />

    <title>Document</title>

    <script type="module" src="../static/register.js" defer></script>

    <!-- <script src="{{url\_for('static', filename = 'register.js')}}"></script> -->

  </head>

  <body>

    <!-- From Uiverse.io by JohnnyCSilva -->

    <div class="main">

      <div class="left">

        <img

          class="login-img"

          src="../static/assets/login-bg.png"

          alt="Background Image"

        />

        <div class="text-overlay">Welcome</div>

        <p class="text-overlay-p">

          Hi, Signup to create account on SmartDentAi.

        </p>

      </div>

      <form class="form">

        <div id="signUpMessage" class="messsageDiv" style="display: none"></div>

        <h2 class="name-login">

          Smart<span class="span-name-login">Dent</span>Ai

        </h2>

        <!-- Added Name Field -->

        <div class="flex-column">

          <label>Name </label>

        </div>

        <div class="inputForm">

          <svg

            xmlns="http://www.w3.org/2000/svg"

            width="20"

            viewBox="0 0 32 32"

            height="20"

          >

            <g data-name="Layer 3" id="Layer\_3">

              <path

                d="m30.853 13.87a15 15 0 0 0 -29.729 4.082 15.1 15.1 0 0 0 12.876 12.918 15.6 15.6 0 0 0 2.016.13 14.85 14.85 0 0 0 7.715-2.145 1 1 0 1 0 -1.031-1.711 13.007 13.007 0 1 1 5.458-6.529 2.149 2.149 0 0 1 -4.158-.759v-10.856a1 1 0 0 0 -2 0v1.726a8 8 0 1 0 .2 10.325 4.135 4.135 0 0 0 7.83.274 15.2 15.2 0 0 0 .823-7.455zm-14.853 8.13a6 6 0 1 1 6-6 6.006 6.006 0 0 1 -6 6z"

              ></path>

            </g>

          </svg>

          <input

            placeholder="Enter your Name"

            class="input"

            type="text"

            id="name"

            required

          />

        </div>

        <!-- End of Added Name Field -->

        <div class="flex-column">

          <label>Email </label>

        </div>

        <div class="inputForm">

          <svg

            xmlns="http://www.w3.org/2000/svg"

            width="20"

            viewBox="0 0 32 32"

            height="20"

          >

            <g data-name="Layer 3" id="Layer\_3">

              <path

                d="m30.853 13.87a15 15 0 0 0 -29.729 4.082 15.1 15.1 0 0 0 12.876 12.918 15.6 15.6 0 0 0 2.016.13 14.85 14.85 0 0 0 7.715-2.145 1 1 0 1 0 -1.031-1.711 13.007 13.007 0 1 1 5.458-6.529 2.149 2.149 0 0 1 -4.158-.759v-10.856a1 1 0 0 0 -2 0v1.726a8 8 0 1 0 .2 10.325 4.135 4.135 0 0 0 7.83.274 15.2 15.2 0 0 0 .823-7.455zm-14.853 8.13a6 6 0 1 1 6-6 6.006 6.006 0 0 1 -6 6z"

              ></path>

            </g>

          </svg>

          <input

            placeholder="Enter your Email"

            class="input"

            type="text"

            id="email"

            required

          />

        </div>

        <div class="flex-column">

          <label>Password </label>

        </div>

        <div class="inputForm">

          <svg

            xmlns="http://www.w3.org/2000/svg"

            width="20"

            viewBox="-64 0 512 512"

            height="20"

          >

            <path

              d="m336 512h-288c-26.453125 0-48-21.523438-48-48v-224c0-26.476562 21.546875-48 48-48h288c26.453125 0 48 21.523438 48 48v224c0 26.476562-21.546875 48-48 48zm-288-288c-8.8125 0-16 7.167969-16 16v224c0 8.832031 7.1875 16 16 16h288c8.8125 0 16-7.167969 16-16v-224c0-8.832031-7.1875-16-16-16zm0 0"

            ></path>

            <path

              d="m304 224c-8.832031 0-16-7.167969-16-16v-80c0-52.929688-43.070312-96-96-96s-96 43.070312-96 96v80c0 8.832031-7.167969 16-16 16s-16-7.167969-16-16v-80c0-70.59375 57.40625-128 128-128s128 57.40625 128 128v80c0 8.832031-7.167969 16-16 16zm0 0"

            ></path>

          </svg>

          <input

            placeholder="Enter your Password"

            class="input"

            type="password"

            id="password"

            required

          />

        </div>

        <div class="flex-row">

          <div>

            <input type="radio" />

            <label>Remember me </label>

          </div>

          <!-- <span class="span">Forgot password?</span> -->

        </div>

        <button

          class="button-submit"

          id="submit"

          type="submit"

          name="action"

          value="login.html"

        >

          Sign Up

        </button>

        <p class="p">

          Do have an account?

          <span class="span">

            <a href="{{ url\_for('login') }}" class="span">Sign In</a></span

          >

        </p>

        <p class="p line">Or With</p>

        <div class="flex-row">

          <button class="btn google">

            <svg

              xml:space="preserve"

              style="enable-background: new 0 0 512 512"

              viewBox="0 0 512 512"

              y="0px"

              x="0px"

              xmlns:xlink="http://www.w3.org/1999/xlink"

              xmlns="http://www.w3.org/2000/svg"

              id="Layer\_1"

              width="20"

              version="1.1"

            >

              <path

                d="M113.47,309.408L95.648,375.94l-65.139,1.378C11.042,341.211,0,299.9,0,256

  c0-42.451,10.324-82.483,28.624-117.732h0.014l57.992,10.632l25.404,57.644c-5.317,15.501-8.215,32.141-8.215,49.456

  C103.821,274.792,107.225,292.797,113.47,309.408z"

                style="fill: #fbbb00"

              ></path>

              <path

                d="M507.527,208.176C510.467,223.662,512,239.655,512,256c0,18.328-1.927,36.206-5.598,53.451

  c-12.462,58.683-45.025,109.925-90.134,146.187l-0.014-0.014l-73.044-3.727l-10.338-64.535

  c29.932-17.554,53.324-45.025,65.646-77.911h-136.89V208.176h138.887L507.527,208.176L507.527,208.176z"

                style="fill: #518ef8"

              ></path>

              <path

                d="M416.253,455.624l0.014,0.014C372.396,490.901,316.666,512,256,512

  c-97.491,0-182.252-54.491-225.491-134.681l82.961-67.91c21.619,57.698,77.278,98.771,142.53,98.771

  c28.047,0,54.323-7.582,76.87-20.818L416.253,455.624z"

                style="fill: #28b446"

              ></path>

              <path

                d="M419.404,58.936l-82.933,67.896c-23.335-14.586-50.919-23.012-80.471-23.012

  c-66.729,0-123.429,42.957-143.965,102.724l-83.397-68.276h-0.014C71.23,56.123,157.06,0,256,0

  C318.115,0,375.068,22.126,419.404,58.936z"

                style="fill: #f14336"

              ></path>

            </svg>

            Google</button

          ><button class="btn apple">

            <svg

              xml:space="preserve"

              style="enable-background: new 0 0 22.773 22.773"

              viewBox="0 0 22.773 22.773"

              y="0px"

              x="0px"

              xmlns:xlink="http://www.w3.org/1999/xlink"

              xmlns="http://www.w3.org/2000/svg"

              id="Capa\_1"

              width="20"

              height="20"

              version="1.1"

            >

              <g>

                <g>

                  <path

                    d="M15.769,0c0.053,0,0.106,0,0.162,0c0.13,1.606-0.483,2.806-1.228,3.675c-0.731,0.863-1.732,1.7-3.351,1.573 c-0.108-1.583,0.506-2.694,1.25-3.561C13.292,0.879,14.557,0.16,15.769,0z"

                  ></path>

                  <path

                    d="M20.67,16.716c0,0.016,0,0.03,0,0.045c-0.455,1.378-1.104,2.559-1.896,3.655c-0.723,0.995-1.609,2.334-3.191,2.334 c-1.367,0-2.275-0.879-3.676-0.903c-1.482-0.024-2.297,0.735-3.652,0.926c-0.155,0-0.31,0-0.462,0 c-0.995-0.144-1.798-0.932-2.383-1.642c-1.725-2.098-3.058-4.808-3.306-8.276c0-0.34,0-0.679,0-1.019 c0.105-2.482,1.311-4.5,2.914-5.478c0.846-0.52,2.009-0.963,3.304-0.765c0.555,0.086,1.122,0.276,1.619,0.464 c0.471,0.181,1.06,0.502,1.618,0.485c0.378-0.011,0.754-0.208,1.135-0.347c1.116-0.403,2.21-0.865,3.652-0.648 c1.733,0.262,2.963,1.032,3.723,2.22c-1.466,0.933-2.625,2.339-2.427,4.74C17.818,14.688,19.086,15.964,20.67,16.716z"

                  ></path>

                </g>

              </g>

            </svg>

            Apple

          </button>

        </div>

      </form>

    </div>

  </body>

</html>

**templates/user.html:**

**user.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Profile</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

    <link rel="stylesheet" href="../static/styles.css" />

    <script type="module" src="../static/profile.js" defer></script>

</head>

<body>

    <header class="header">

        <nav>

            <ul class="head2">

                <li class="heads">Smart<span class="main-logo">Dent</span>Ai</li>

            </ul>

            <ul class="head">

                <a href="{{ url\_for('home') }}"><li class="heads">🏠</li></a>

                <a href="{{ url\_for('contact') }}"><li class="heads">📞</li></a>

                <a href="{{ url\_for('about') }}"><li class="heads">ℹ️</li></a>

            </ul>

            <ul class="head1">

                <button class="btn btn-outline-primary" id="logout">Sign out</button>

            </ul>

        </nav>

    </header>

    <div class="container py-5">

        <div class="card mb-3">

            <div class="card-body">

                <h4 class="card-title">User Profile</h4>

                <p><strong>Name:</strong> <span id="username"></span></p>

                <p><strong>Email:</strong> <span id="email"></span></p>

                <p><strong>Profile Picture:</strong></p>

                <img src="{{ url\_for('static', filename='assets/default-profile.png') }}" alt="Profile Picture" id="userProfilePicture" class="img-fluid rounded-circle mb-3" style="width: 100px; height: 100px;" />

                <div class="mt-3">

                    <input type="file" id="profilePictureInput" accept="image/\*" class="form-control mb-2" />

                    <button id="uploadProfilePicture" class="btn btn-primary">Upload Profile Picture</button>

                </div>

            </div>

        </div>

        <!-- User History Section -->

        <div class="card">

            <div class="card-body">

                <h4 class="card-title">User History</h4>

                <table class="table table-striped">

                    <thead>

                        <tr>

                            <th>Sr. No.</th>

                            <th>Input Image</th>

                            <th>Output Result</th>

                        </tr>

                    </thead>

                    <tbody id="userHistory">

                        <!-- Example Row -->

                        <tr>

                            <td>1</td>

                            <td>

                                <img src="path/to/input-image.jpg" alt="Input Preview" class="img-thumbnail" style="width: 100px; height: 100px;" />

                                <br>

                                <a href="path/to/input-image.jpg" download class="btn btn-sm btn-outline-primary mt-2">Download</a>

                            </td>

                            <td>

                                <img src="path/to/output-image.jpg" alt="Output Preview" class="img-thumbnail" style="width: 100px; height: 100px;" onerror="this.style.display='none';" />

                                <span id="outputText">Output Text Example</span>

                                <br>

                                <a href="path/to/output-image.jpg" download class="btn btn-sm btn-outline-primary mt-2">Download</a>

                            </td>

                        </tr>

                    </tbody>

                </table>

            </div>

        </div>

    </div>

</body>

</html>

**.env:**

GOOGLE\_API\_KEY='AIzaSyDggauQaeKzjEcWDnPENqNg0C8LqIOh6yI'

# 📁 .env -----

AUTH0\_CLIENT\_ID=DlCdee6zGwbRQ5voFEdBx9aLWjnQlO6M

AUTH0\_CLIENT\_SECRET=gu\_HDbC-85MOUBStVBeVnzYwuksXccBtdEMaCiJv6HPvmoD5OXbHoH8kczCI6NxC

AUTH0\_DOMAIN=dev-fqhxkfbz40qerm43.us.auth0.com

APP\_SECRET\_KEY=

**App.py**

import os

import traceback

import sqlite3

import time

from flask import Flask, request, render\_template, redirect, url\_for, jsonify

from flask\_cors import CORS  # Added for CORS support

from tensorflow.keras.models import load\_model

import google.generativeai as genai

from google.generativeai.types import HarmCategory, HarmBlockThreshold

import tensorflow as tf

from werkzeug.utils import secure\_filename

import numpy as np

from PIL import Image

import torch

import torch.nn as nn

import webbrowser

import sys

import subprocess

from dotenv import load\_dotenv

# from oral import OralDiseaseModel   #import the class from oral.py

# Load environment variables from .env file

load\_dotenv()

app = Flask(\_\_name\_\_)

# Apply CORS globally to catch any misconfigured requests

CORS(app, resources={r"/\*": {"origins": "http://localhost:5173"}})

UPLOAD\_FOLDER = 'static/assets/uploads'

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER  # Set upload folder in app config

# MODEL\_PATH\_XRAY = 'model/Xray\_Pred\_model.h5'

# MODEL\_PATH\_DENTAL = 'oral\_disease\_pred/model/oral\_disease\_best\_model.h5'

# model\_xray = load\_model(MODEL\_PATH\_XRAY)

# model\_dental = load\_model(MODEL\_PATH\_DENTAL)

# Ensure API key is set

api\_key = os.getenv("GOOGLE\_API\_KEY")

if not api\_key:

    raise ValueError("GOOGLE\_API\_KEY is not set in environment variables.")

genai.configure(api\_key=os.getenv("GOOGLE\_API\_KEY"))

# Image preprocessing function

IMG\_SIZE = (256, 256)

def preprocess\_image(img\_path):

    img = Image.open(img\_path)

    img = img.resize(IMG\_SIZE)

    img = np.array(img) / 255.0  # Normalize the image

    img = np.expand\_dims(img, axis=0)  # Add batch dimension

    return img

# Allowed file extensions for upload

def allowed\_file(filename):

    ALLOWED\_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

# Main page is for login and signup button

@app.route('/')

def main():

    return render\_template('main.html')

@app.route('/home', methods=['GET', 'POST'])

def home():

    return render\_template('index.html')

@app.route('/back\_btn', methods=['GET', 'POST'])

def back\_btn():

    return redirect(url\_for('home'))

# Redirecting to the signup page

@app.route('/start', methods=['GET', 'POST'])

def start():

    print("Start route triggered")

    return redirect(url\_for('signup'))

@app.route('/login', methods=['GET', 'POST'])

def login():

    return render\_template('login.html')

@app.route('/contact', methods=['GET', 'POST'])

def contact():

    return render\_template('contact\_us.html')

@app.route('/about', methods=['GET', 'POST'])

def about():

    return render\_template('about\_us.html')

@app.route('/profile', methods=['GET', 'POST'])

def profile():

    return render\_template('profile.html')

@app.route('/signup', methods=['GET', 'POST'])

def signup():

    if request.method == 'POST':

        email = request.form.get('email')

        password = request.form.get('password')

        print(f"Signup attempt: Email={email}, Password={password}")

        return redirect(url\_for('login'))  # Redirect to login page after signup

    return render\_template('Signup.html')  # For GET requests, render the signup page

MODEL\_PATH\_XRAY = 'model/Xray\_Pred\_model.h5'

MODEL\_PATH\_DENTAL = 'C:\MajorProject\SmartDentAi-project\Oral\model\oral\_disease\_best\_model.h5'

model\_xray = load\_model(MODEL\_PATH\_XRAY)

model\_dental = load\_model(MODEL\_PATH\_DENTAL)

# Route to handle file upload, prediction, and display everything in one page

@app.route('/predict', methods=['POST'])

def predict():

    if 'file' not in request.files:

        return redirect(url\_for('home'))  # Changed from 'index' to 'home' to match your routes

    file = request.files['file']

    if file and allowed\_file(file.filename):

        filename = secure\_filename(file.filename)

        file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

        file.save(file\_path)

        # Preprocess the image with the correct target size

        img = preprocess\_image(file\_path)

        if file:

            img = Image.open(file).convert('RGB')  # Ensure 3-channel image

            img = img.resize((224, 224))  # Resize to model's required shape

            img = np.array(img) / 255.0  # Normalize pixel values

            img = np.expand\_dims(img, axis=0)  # Add batch dimension

            # Ensure input shape is correct

            print("Processed Image Shape:", img.shape)  # Debugging

        # Make prediction

        prediction = model\_xray.predict(img)

        predicted\_class = np.argmax(prediction)

        # Class mapping

        classes = {0: 'Cavity', 1: 'Fillings', 2: 'Impacted Tooth', 3: 'Implant'}

        result = classes[predicted\_class]

        # Diagnosis information for each predicted class

        diagnosis\_info = {

            'Fillings': {

                'symptoms': 'Toothache, sensitivity, pain while chewing, visible holes in teeth.',

                'causes': 'Tooth decay due to plaque build-up and bacterial infections.',

                'treatment': 'The dentist removes decayed material and fills the cavity with a composite material.',

                'precautions': 'Brush and floss daily, avoid sugary foods, and visit the dentist regularly.'

            },

            'Implant': {

                'symptoms': 'Missing tooth, discomfort while eating or speaking.',

                'causes': 'Tooth loss due to decay, injury, or periodontal disease.',

                'treatment': 'A dental implant is surgically placed in the jawbone and acts as a root for a replacement tooth.',

                'precautions': 'Good oral hygiene, avoid smoking, and follow the dentist’s aftercare instructions.'

            },

            'Impacted Tooth': {

                'symptoms': 'Swelling, pain, difficulty opening mouth, bad breath.',

                'causes': 'A tooth that fails to fully erupt due to lack of space or alignment issues.',

                'treatment': 'May require surgical removal or orthodontic treatment.',

                'precautions': 'Regular dental check-ups, good oral hygiene, and timely intervention for wisdom teeth.'

            },

            'Cavity': {

                'symptoms': 'Toothache, sensitivity to hot/cold, visible holes in teeth, bad breath.',

                'causes': 'Plaque build-up leading to acid erosion and tooth decay.',

                'treatment': 'Fillings, crowns, or in severe cases, root canal treatment.',

                'precautions': 'Brush with fluoride toothpaste, reduce sugary food intake, and visit the dentist regularly.'

            }

        }

        # Retrieve diagnosis information for the predicted class

        details = diagnosis\_info.get(result, {})

        return render\_template('predict.html', prediction=result, details=details, img\_path='assets/uploads/' + filename)

    return redirect(url\_for('home'))

# Route to handle file upload, prediction, and display everything in one page

@app.route('/predict2', methods=['POST'])

def predict2():

    if 'file' not in request.files:

        return redirect(url\_for('home'))  # Changed from 'index' to 'home' to match your routes

    file = request.files['file']

    if file and allowed\_file(file.filename):

        filename = secure\_filename(file.filename)

        file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

        file.save(file\_path)

        # Preprocess the image with the correct target size

        img = preprocess\_image(file\_path)

        if file:

            img = Image.open(file).convert('RGB')  # Ensure 3-channel image

            img = img.resize((224, 224))  # Resize to model's required shape

            img = np.array(img) / 255.0  # Normalize pixel values

            img = np.expand\_dims(img, axis=0)  # Add batch dimension

            # Ensure input shape is correct

            print("Processed Image Shape:", img.shape)  # Debugging

        # Make prediction

        prediction = model\_dental.predict(img)

        predicted\_class = np.argmax(prediction)

        # Class mapping

        classes = {0: 'Fillings', 1: 'Implant', 2: 'Impacted Tooth', 3: 'Cavity'}

        result = classes[predicted\_class]

        # Diagnosis information for each predicted class

        diagnosis\_info = {

            'Fillings': {

                'symptoms': 'Toothache, sensitivity, pain while chewing, visible holes in teeth.',

                'causes': 'Tooth decay due to plaque build-up and bacterial infections.',

                'treatment': 'The dentist removes decayed material and fills the cavity with a composite material.',

                'precautions': 'Brush and floss daily, avoid sugary foods, and visit the dentist regularly.'

            },

            'Implant': {

                'symptoms': 'Missing tooth, discomfort while eating or speaking.',

                'causes': 'Tooth loss due to decay, injury, or periodontal disease.',

                'treatment': 'A dental implant is surgically placed in the jawbone and acts as a root for a replacement tooth.',

                'precautions': 'Good oral hygiene, avoid smoking, and follow the dentist’s aftercare instructions.'

            },

            'Impacted Tooth': {

                'symptoms': 'Swelling, pain, difficulty opening mouth, bad breath.',

                'causes': 'A tooth that fails to fully erupt due to lack of space or alignment issues.',

                'treatment': 'May require surgical removal or orthodontic treatment.',

                'precautions': 'Regular dental check-ups, good oral hygiene, and timely intervention for wisdom teeth.'

            },

            'Cavity': {

                'symptoms': 'Toothache, sensitivity to hot/cold, visible holes in teeth, bad breath.',

                'causes': 'Plaque build-up leading to acid erosion and tooth decay.',

                'treatment': 'Fillings, crowns, or in severe cases, root canal treatment.',

                'precautions': 'Brush with fluoride toothpaste, reduce sugary food intake, and visit the dentist regularly.'

            }

        }

        # Retrieve diagnosis information for the predicted class

        details = diagnosis\_info.get(result, {})

        return render\_template('predict.html', prediction=result, details=details, img\_path='assets/uploads/' + filename)

    return redirect(url\_for('home'))

# @app.route('/ortho\_predict', methods = ['GET', 'POST'])

# def ortho\_predict():

#     return render\_template("orthodontic.html")

@app.route('/ortho\_predict', methods = ['GET', 'POST'])

def ortho\_predict():

    return redirect("http://localhost:5006")

# @app.route('/orthodontic', methods=['POST'])

# def orthodontic():

    # with custom\_app.app\_context():  # Use the second Flask instance

        # return render\_template('index.html')  # Uses the "SmartDentAi-project/edualc/code/templates" folder

# @app.route('/ortho\_predict', methods=['GET','POST'])

# def ortho\_predict():

#     try:

#         script\_path = "../edualc/code/app.py"

#         script\_path = r"C:\MajorProject\SmartDentAi-project\edualc\code\app.py"

#         result = subprocess.run([sys.executable, script\_path], capture\_output=True, text=True)

#         if result.returncode == 0:

#             return jsonify({"message": "Script executed successfully", "output": result.stdout}), 200

#         else:

#             app.logger.error(f"Script Error: {result.stderr}")

#             return jsonify({

#                 "error": "Script execution failed",

#                 "details": result.stderr,

#                 "return\_code": result.returncode

#             }), 500

#     except Exception as e:

#         app.logger.error(f"Exception: {traceback.format\_exc()}")

#         return jsonify({"error": "Internal Server Error", "details": str(e)}), 500

@app.route('/remote', methods = ['GET', 'POST'])

def remote():

    return redirect("http://localhost:5002")

@app.route('/dentalimg', methods = ['GET', 'POST'])

def dentalimg():

    return redirect("http://localhost:5003")

# @app.route('/remote', methods=['GET', 'POST'])

# def run\_remote\_script():

#     script\_path = os.path.abspath(os.path.join(os.getcwd(), 'remote', '../Remote/main.py'))

#     if not os.path.exists(script\_path):

#         return jsonify({'error': f"Script not found at {script\_path}"})

#     try:

#         result = subprocess.run(

#             [sys.executable, script\_path],

#             capture\_output=True, text=True

#         )

#         return jsonify({

#             'output': result.stdout.strip(),

#             'error': result.stderr.strip(),

#             'return\_code': result.returncode

#         })

#     except Exception as e:

#         return jsonify({'error': str(e)})

# Path to your React Vite project

REACT\_VITE\_PATH = r"C:\MajorProject\SmartDentAi-project\Appointment"

NPM\_PATH = r"C:\Program Files\nodejs\npm.cmd"

@app.route('/appointment', methods=['GET', 'POST'])

def appointment():

    try:

        if not os.path.exists(REACT\_VITE\_PATH):

            return jsonify({"error": f"Directory does not exist: {REACT\_VITE\_PATH}"})

        if not os.path.isdir(REACT\_VITE\_PATH):

            return jsonify({"error": f"Path is not a directory: {REACT\_VITE\_PATH}"})

        if not os.path.exists(NPM\_PATH):

            return jsonify({"error": f"npm not found at: {NPM\_PATH}"})

        process = subprocess.Popen(

            [NPM\_PATH, "run", "dev"],

            cwd=REACT\_VITE\_PATH,

            stdout=subprocess.PIPE,

            stderr=subprocess.PIPE,

            shell = True,

            # creationflags=subprocess.CREATE\_NEW\_CONSOLE

        )

        time.sleep(2)

        if process.poll() is not None and process.returncode != 0:

            stderr = process.stderr.read().decode()

            return jsonify({

                "error": "Vite server failed to start",

                "details": stderr if stderr else "Unknown error"

            })

        return redirect("http://localhost:5173")

    except Exception as e:

        return jsonify({"error": str(e)})

# Test route to verify API is reachable

@app.route('/api/test', methods=['GET'])

def test\_api():

    print("Test API endpoint hit")

    return jsonify({"message": "API is reachable"}), 200

# Catch-all route for /api/\* to debug unmatched requests

@app.route('/api/<path:path>', methods=['GET', 'POST', 'PUT', 'DELETE'])

def catch\_all\_api(path):

    print(f"Unmatched API request: /api/{path}, Method: {request.method}, Data: {request.json}")

    return jsonify({"error": f"Endpoint /api/{path} not found"}), 404

@app.route('/api/book\_appointment/<doctor\_id>', methods=['POST'])

def book\_appointment(doctor\_id):

    print(f"Received request for /api/book\_appointment/{doctor\_id}")

    try:

        data = request.json

        print("Received booking data:", data)

        selected\_date = data.get('date')  # e.g., "25\_3\_2025"

        selected\_time = data.get('time')  # e.g., "10:00 AM"

        if not selected\_date or not selected\_time:

            print("Validation failed: Date or time missing")

            return jsonify({"error": "Date and time are required"}), 400

        # Mock doctor data (replace with database query in production)

        doctors = [

            {"\_id": "1", "name": "Dr. Richard James", "speciality": "MBBS - General Dentist", "image": "path/to/image.jpg"},

            {"\_id": "doc3", "name": "Dr. Sarah Smith", "speciality": "Orthodontist", "image": "path/to/image2.jpg"}  # Added doc3

        ]

        doctor = next((doc for doc in doctors if doc["\_id"] == doctor\_id), None)

        if not doctor:

            print(f"Doctor not found for ID: {doctor\_id}")

            return jsonify({"error": "Doctor not found"}), 404

        # Save to database

        try:

            # Ensure the database file is created in the project directory

            db\_path = os.path.join(os.getcwd(), 'bookings.db')

            print(f"Connecting to database at: {db\_path}")

            conn = sqlite3.connect(db\_path)

            cursor = conn.cursor()

            cursor.execute('''CREATE TABLE IF NOT EXISTS bookings

                            (doctor\_id TEXT, doctor\_name TEXT, date TEXT, time TEXT, user\_id TEXT)''')

            # For simplicity, using a static user\_id; in production, get this from the logged-in user

            user\_id = "user1"

            cursor.execute('INSERT INTO bookings (doctor\_id, doctor\_name, date, time, user\_id) VALUES (?, ?, ?, ?, ?)',

                          (doctor\_id, doctor["name"], selected\_date, selected\_time, user\_id))

            conn.commit()

            conn.close()

            print(f"Booking saved for Doctor ID {doctor\_id} on {selected\_date} at {selected\_time}")

        except sqlite3.Error as e:

            print(f"SQLite error: {str(e)}")

            return jsonify({"error": f"Database error: {str(e)}"}), 500

        return jsonify({

            "message": "Appointment booked successfully!",

            "doctor\_name": doctor["name"],

            "date": selected\_date,

            "time": selected\_time,

            "doctor\_id": doctor\_id

        }), 200

    except Exception as e:

        app.logger.error(f"Error in book\_appointment: {traceback.format\_exc()}")

        return jsonify({"error": str(e)}), 500

@app.route('/api/my\_appointments/<user\_id>', methods=['GET'])

def get\_my\_appointments(user\_id):

    print(f"Received request for /api/my\_appointments/{user\_id}")

    try:

        conn = sqlite3.connect('bookings.db')

        cursor = conn.cursor()

        cursor.execute('SELECT doctor\_id, doctor\_name, date, time FROM bookings WHERE user\_id = ?', (user\_id,))

        appointments = cursor.fetchall()

        conn.close()

        # Format the response

        appointments\_list = [

            {"doctor\_id": appt[0], "doctor\_name": appt[1], "date": appt[2], "time": appt[3]}

            for appt in appointments

        ]

        print(f"Fetched appointments for user {user\_id}: {appointments\_list}")

        return jsonify(appointments\_list), 200

    except Exception as e:

        app.logger.error(f"Error in get\_my\_appointments: {traceback.format\_exc()}")

        return jsonify({"error": str(e)}), 500

@app.route('/api/doctors/<doctor\_id>', methods=['GET'])

def get\_doctor(doctor\_id):

    print(f"Received request for /api/doctors/{doctor\_id}")

    try:

        # Mock data; replace with database query

        doctors = [

            {"\_id": "1", "name": "Dr. Richard James", "speciality": "MBBS - General Dentist", "image": "path/to/image.jpg"},

            {"\_id": "doc3", "name": "Dr. Sarah Smith", "speciality": "Orthodontist", "image": "path/to/image2.jpg"}  # Added doc3

        ]

        doctor = next((doc for doc in doctors if doc["\_id"] == doctor\_id), None)

        if doctor:

            return jsonify(doctor), 200

        return jsonify({"error": "Doctor not found"}), 404

    except Exception as e:

        app.logger.error(f"Error in get\_doctor: {traceback.format\_exc()}")

        return jsonify({"error": str(e)}), 500

@app.route('/api', methods=['POST'])

def api\_endpoint():

    print("Received request for /api")

    try:

        data = request.json

        print("Received Data:", data)

        user\_input = data.get('message', '')

        if not user\_input:

            return jsonify({"error": "No input provided"}), 400

        model = genai.GenerativeModel("gemini-pro")

        generationConfig = {

            "temperature": 0.9,

            "topK": 1,

            "topP": 1,

            "maxOutputTokens": 1000,

        }

        response = model.generate\_content(user\_input)

        print("Generated Response:", response)

        return jsonify({"response": response.text})

    except Exception as e:

        app.logger.error(f"Error in api\_endpoint: {traceback.format\_exc()}")

        return jsonify({"error": str(e)}), 500

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True, port=5001, use\_reloader = False, threaded = True)

**requirements.txt:**

Flask

tensorflow

numpy

Pillow

# 📁 requirements.txt -----

flask>=2.0.3

python-dotenv>=0.19.2

authlib>=1.0

requests>=2.27.1

**server.py:**

# 📁 server.py -----

import json

from os import environ as env

from urllib.parse import quote\_plus, urlencode

from authlib.integrations.flask\_client import OAuth

from dotenv import find\_dotenv, load\_dotenv

from flask import Flask, redirect, render\_template, session, url\_for

ENV\_FILE = find\_dotenv()

if ENV\_FILE:

    load\_dotenv(ENV\_FILE)

# 👆 We're continuing from the steps above. Append this to your server.py file.

app = Flask(\_\_name\_\_)

app.secret\_key = env.get("APP\_SECRET\_KEY")

# 👆 We're continuing from the steps above. Append this to your server.py file.

oauth = OAuth(app)

oauth.register(

    "auth0",

    client\_id=env.get("AUTH0\_CLIENT\_ID"),

    client\_secret=env.get("AUTH0\_CLIENT\_SECRET"),

    client\_kwargs={

        "scope": "openid profile email",

    },

    server\_metadata\_url=f'https://{env.get("AUTH0\_DOMAIN")}/.well-known/openid-configuration'

)

# 👆 We're continuing from the steps above. Append this to your server.py file.

@app.route("/login")

def login():

    return oauth.auth0.authorize\_redirect(

        redirect\_uri=url\_for("callback", \_external=True)

    )

# 👆 We're continuing from the steps above. Append this to your server.py file.

@app.route("/callback", methods=["GET", "POST"])

def callback():

    token = oauth.auth0.authorize\_access\_token()

    session["user"] = token

    return redirect("/")

# 👆 We're continuing from the steps above. Append this to your server.py file.

@app.route("/logout")

def logout():

    session.clear()

    return redirect(

        "https://" + env.get("AUTH0\_DOMAIN")

        + "/v2/logout?"

        + urlencode(

            {

                "returnTo": url\_for("home", \_external=True),

                "client\_id": env.get("AUTH0\_CLIENT\_ID"),

            },

            quote\_via=quote\_plus,

        )

    )

# 👆 We're continuing from the steps above. Append this to your server.py file.

@app.route("/")

def home():

    return render\_template("home.html", session=session.get('user'), pretty=json.dumps(session.get('user'), indent=4))

# 👆 We're continuing from the steps above. Append this to your server.py file.

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host="0.0.0.0", port=env.get("PORT", 3000))

**test\_api.py:**

import requests

url = "http://127.0.0.1:5001/api"

data = {"message": "Hello"}

response = requests.post(url, json=data)

print(f"Status Code: {response.status\_code}")

print(f"Response: {response.json()}")

**test\_your\_file.py:**

from app import app  # Import your Flask app

app.testing = True

client = app.test\_client()

def test\_api():

    response = client.post('/api', json={'message': 'Hello'})

    assert response.status\_code == 200

    assert response.json == {"response": "Expected response"}

if \_\_name\_\_ == '\_\_main\_\_':

    test\_api()