npm install firebase

composer require kreait/firebase-php

composer show laravel-notification-channels/fcm

php artisan vendor:publish --provider="NotificationChannels\Fcm\FcmServiceProvider"

pulibic/firebase-messaging-sw.js :note that import.meta.env does not work here

importScripts('https://www.gstatic.com/firebasejs/10.12.0/firebase-app-compat.js');

importScripts('https://www.gstatic.com/firebasejs/10.12.0/firebase-messaging-compat.js');

const firebaseConfig = {

    apiKey: "",

    authDomain: "",

    projectId: "",

    storageBucket: "",

    messagingSenderId: "",

    appId: “"

};

firebase.initializeApp(firebaseConfig);

const messaging = firebase.messaging();

messaging.onBackgroundMessage(function(payload) {

  console.log('[firebase-messaging-sw.js] Received background message ', payload);

  const notificationTitle = payload.notification.title;

  const notificationOptions = {

    body: payload.notification.body,

    icon: '/favicon.ico',

    data: {

      click\_action: payload.data?.click\_action || '/'

    }

  };

  self.registration.showNotification(notificationTitle, notificationOptions);

});

self.addEventListener('notificationclick', function(event) {

  event.notification.close();

  const clickAction = event.notification.data?.click\_action;

  if (clickAction) {

    event.waitUntil(clients.openWindow(clickAction));

  }

});

hooks/use-fcm-token.ts :

import { useEffect } from 'react';

import { initializeApp } from "firebase/app";

import { getMessaging, getToken } from "firebase/messaging";

import { usePage, router } from '@inertiajs/react';

import { SharedData } from '@/types';

const firebaseConfig = {

    apiKey: "",

    authDomain: "",

    projectId: "",

    storageBucket: "",

    messagingSenderId: “",

    appId: ""

};

const app = initializeApp(firebaseConfig);

const messaging = getMessaging(app);

export function useFcmToken() {

    const csrf = usePage<SharedData>().props.csrf\_token;

    useEffect(() => {

        getToken(messaging, { vapidKey: "your\_vapid\_key" })

            .then((currentToken) => {

                if (currentToken) {

                    fetch('/store-device-token', {

                        method: 'POST',

                        headers: {

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

                            'X-CSRF-TOKEN': csrf

                        },

                        body: JSON.stringify({ token: currentToken }),

                        credentials: 'same-origin'

                    });

                }

            }).catch((err) => {

                console.log('Error getting token:', err);

            });

    }, []);

}

Download the firebase.json and add it to /strorage/app/firebase

Services/Firebase/FirebaseService

<?php

namespace App\Services\Firebase;

use Kreait\Firebase\Factory;

use Kreait\Firebase\Messaging\CloudMessage;

use Kreait\Firebase\Messaging\Notification;

class FirebaseService

{

    protected $messaging;

    public function \_\_construct()

    {

        $factory = (new Factory)->withServiceAccount(storage\_path('app/firebase/firebase\_credentials.json'));

        $this->messaging = $factory->createMessaging();

    }

    public function subscribeToTopic(string $token, string $topic)

    {

        $this->messaging->subscribeToTopic($topic, [$token]);

    }

    public function sendToTopic(string $topic, string $title, string $body)

    {

        $message = CloudMessage::withTarget('topic', $topic)

            ->withNotification(Notification::create($title, $body));

        return $this->messaging->send($message);

    }

}

web.php

    Route::post('/store-device-token', [FcmController::class, 'storeToken']);

    Route::post('/send-topic-notification', [FcmController::class, 'sendToTopic']);

FCMCONTROLLER

<?php

namespace App\Http\Controllers;

use App\Models\User;

use App\Services\Firebase\FirebaseService;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class FcmController extends Controller

{

    protected $firebase;

    public function \_\_construct(FirebaseService $firebase)

    {

        $this->firebase = $firebase;

    }

    public function storeToken(Request $request)

    {

        $request->validate(['token' => 'required|string']);

        /\*\*

         \* @var User $user

         \*/

        $user = Auth::user();

        $user->fcmTokens()->updateOrCreate([

            'token' => $request->token,

        ]);

        // Subscribe to topic automatically

        $this->firebase->subscribeToTopic($request->token, 'all\_users');

        return response()->json(['message' => 'Token stored & subscribed']);

    }

    public function sendToTopic(Request $request)

    {

        $request->validate([

            'title' => 'required|string',

            'body' => 'required|string',

        ]);

        $this->firebase->sendToTopic('all\_users', $request->title, $request->body);

        // $user = User::find(1);

        // $user->notify(new PushNotification());

        return response()->json(['message' => 'Notification sent']);

    }

}

Fcm push notification

<?php

namespace App\Notifications;

use Illuminate\Bus\Queueable;

use Illuminate\Notifications\Notification;

use NotificationChannels\Fcm\FcmChannel;

use NotificationChannels\Fcm\FcmMessage;

use NotificationChannels\Fcm\Resources\Notification as FcmNotification;

class GuestPushNotification extends Notification

{

    use Queueable;

    public function \_\_construct(public $name)

    {

        //

    }

    public function via($notifiable)

    {

        return [FcmChannel::class];

    }

    public function toFcm($notifiable)

    {

        return new FcmMessage(

            notification: new FcmNotification(

                title: 'ACTS Message Notification!',

                body: "{$this->name} has sent a message to the page",

                image: '/images/facility.jpg'

            ),

            data: [

                'click\_action' => route('messages.index'),

                'custom\_data' => 'my\_value',

            ]

        );

    }

}

2025\_06\_20\_133536\_create\_fcm\_tokens\_table.php

 Schema::create('fcm\_tokens', function (Blueprint $table) {

            $table->id();

            $table->foreignId('user\_id')->constrained()->onDelete('cascade');

            $table->string('token')->unique();

            $table->timestamps();

        });

FCMToke (Model)

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class FcmToken extends Model

{

    //

    protected $guarded = [];

    public function user()

    {

        return $this->belongsTo(User::class);

    }

}

User.php

    public function fcmTokens()

    {

        return $this->hasMany(FcmToken::class);

    }

    public function routeNotificationForFcm()

    {

        return $this->fcmTokens()->pluck('token')->toArray();

    }

storage/app/firebase/firebase\_credentials.json

{

  "type": "service\_accouasdasdasdnt",

  "project\_id": "2131231-2",

  "private\_key\_id": "123",

  "private\_key": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCPBkYxnBysjdZL\nV/KDebYfu6guJDd0NOCU6XsLF1xpqzKXvfGqvX40QBRdYS1KB1PBMh3Ov8/A0Crc\n1+mk30mvFI27g0FcWPIVndAA6HiBbt123123HpgpJ8XO7lZ1jcBPI4UxyJ4bEZp8C6fjVB\nK9xrLCwom2fmz93F/XMJj91hDFSVyW4aEuEAGJ50xKoSW6f2AE59m5cfbqSStdws\nfcrsRriurqBojlkNrnML0Cw1HmkNANlD4RtShpbdFrORSwP5e5FpQd8EkPznqqzh\nwEPsadETuuU4aNbebiY9Ng5lxB0sLjMVLQCcfLO2qAdHP3xb4QMUJLZWL+i6AowRn+UGJja1E1feB\n1D1O5RniTf49V/Ef4/IccjGWBR662wi21WlFtbjevSypGVss2/wAn2MmmG4rTFey\nT6tA6DVPsMUhAn2D8PavxJnZ\n-----END PRIVATE KEY-----\n",

  "client\_email": "firebase-admins213dk-fbsvc@acts-dcb8f.iam.gse12acco123unt.com",

  "client\_id": "11123sdfdf607asd00",

  "auth\_uri": "https://accou32asdadh2/auth",

  "token\_uri": "https://oauth2.googleapis.com/token",

  "auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

  "client\_x509\_cert\_url": "https://wwasdasdbase-adminsdk-fbsvc%40acts-dcb8f.iam.gserviceaccount.com",

  "universe\_domain": "googleapis.com"

}

app-layout

import AppLayoutTemplate from '@/layouts/app/app-header-layout';

import { SharedData, type BreadcrumbItem } from '@/types';

import { type ReactNode } from 'react';

import { Toaster } from "@/components/ui/sonner"

import { usePage } from '@inertiajs/react';

import { useFcmToken } from '@/hooks/use-fcm-token';

interface AppLayoutProps {

    children: ReactNode;

    breadcrumbs?: BreadcrumbItem[];

}

// /npm i react-facebook

export default function ({ children, breadcrumbs, ...props }: AppLayoutProps) {

    const { auth } = usePage<SharedData>().props;

    if (auth?.user) {

        useFcmToken();

    }

    return (<AppLayoutTemplate breadcrumbs={breadcrumbs} {...props}>

        {children}

        <Toaster closeButton theme='light' richColors

        />

    </AppLayoutTemplate>);

};

.env

FIREBASE\_PROJECT=app

FIREBASE\_CREDENTIALS=storage/app/firebase/firebase\_credentials.json