Here’s the “admin-approved” Web-Push solution I outlined earlier, repeated in one place:

1. Create VAPID keys

```bash

npx web-push generate-vapid-keys

```

• Save the public key in an environment variable `VAPID\_PUBLIC`.

• Save the private key in `VAPID\_PRIVATE` (never ship it to the client).

2. Add a service-worker (`/public/sw.js`)

```js

self.addEventListener('push', (event) => {

const data = event.data?.json() || {};

event.waitUntil(

self.registration.showNotification(data.title || 'Mareeba Badminton', {

body: data.body,

icon: '/icon.png',

tag: data.tag || 'mbc-general'

})

);

});

```

3. Build a “Notifications” section on `/payments`

```ts

// /payments/page.tsx (or a component)

const subscribe = async () => {

if (Notification.permission !== 'granted') {

await Notification.requestPermission();

}

const reg = await navigator.serviceWorker.ready;

const sub = await reg.pushManager.subscribe({

userVisibleOnly: true,

applicationServerKey: VAPID\_PUBLIC

});

// POST to /api/subscribe

await fetch('/api/subscribe', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify(sub)

});

alert('Subscription created – pending admin approval.');

};

```

4. Store the subscription in Supabase

Table `push\_subscriptions`

• id (uuid)

• endpoint, p256dh, auth (text)

• player\_id (text)

• approved (boolean, default `false`)

• created\_at, approved\_at

5. Build a tiny admin screen (extend your `/admin` page)

• List subscriptions where `approved = false`

• Approve button sets `approved = true`

6. Send notifications from the server (Node / Edge Function)

```ts

import webpush from 'web-push';

webpush.setVapidDetails(

'mailto:you@example.com',

process.env.VAPID\_PUBLIC!,

process.env.VAPID\_PRIVATE!

);

export const sendToApproved = async (payload: any) => {

const { data: subs } = await supabase

.from('push\_subscriptions')

.select('\*')

.eq('approved', true);

for (const s of subs) {

try {

await webpush.sendNotification(s, JSON.stringify(payload));

} catch (err: any) {

if (err.statusCode === 404 || err.statusCode === 410) {

// endpoint gone – delete row

await supabase.from('push\_subscriptions').delete().eq('id', s.id);

}

}

}

};

```

7. Typical payloads you might send

```ts

await sendToApproved({

title: 'Payment confirmed!',

body: `Player ${playerId} just paid.`,

tag: 'payment-' + bookingId

});

```

8. UX flow

• Player visits `/payments` → clicks “Enable Notifications” → gets “Pending admin approval” toast.

• Admin logs in, approves the subscription.

• All future server events (`sendToApproved`) appear on that device.

• A “Stop notifications” button can call `subscription.unsubscribe()` and DELETE the row.

That’s the full opt-in with admin-approval pattern—only four devices, but scalable if you add more later.