

Niaga Prestasi — property & listing platform

Client group: Niaga Prestasi (Malaysia) **Stack:** Laravel · Vue.js · Pusher **Related:** TLMS logistics · RedDot payments

Full-stack property buy/sell with governed CRUD, validation at the API boundary, and Pusher-driven live UI — operators see listing and transaction state change without polling.

LaravelVue.jsPusherCRUD APIsReal-time UI
Architecture

API-first + reactive front end

Laravel owns models, validation rules, and admin CRUD. Vue SPA consumes versioned JSON. Domain events (listing published, offer accepted) broadcast on private Pusher channels scoped per operator role.

Design decisions

Why Pusher on this stack

Option	Trade-off	Chosen
Pusher channels	External dependency	Fast operator UX without running our own WS cluster
Laravel events	Must stay thin	Emit after DB commit — UI never sees rolled-back state
Vue per workflow	More bundles	Buy vs sell vs admin isolated; shared form components
Code shape		

Event after commit (representative)

```
DB::transaction(function () use ($listing) {  
    $listing->transitionTo('published');  
});  
event(new ListingPublished($listing)); // → Pusher private channel
```

Outcomes

Results

- End-to-end listing and transaction UI on Laravel + Vue
- Real-time operator feedback via Pusher
- Delivered alongside TLMS and PayU programmes for same client group

Niaga Prestasi programme. Same client group as TLMS case study.

contact@sevendyne.com · sevendyne.com/case-studies © 2026 Sevendyne Consultancy Services LLP