

# **Wheels & Wins – Pam 2.0 Test Playbook** **(Gemini-First Edition)**

This playbook contains **step-by-step testing prompts** to validate each phase of the Pam 2.0 build. Use after completing the corresponding phase in the **Build Playbook**.

---

## **Phase 1 – Setup & Scaffolding**

### **Test 1.1 – Repo & Branch**

List all branches. Confirm `pam-2.0` exists. Confirm backend directory is new and clean, frontend PAM UI remains intact.

### **Test 1.2 – Health Check**

Start FastAPI server. Call `/health`. Confirm response = {"status": "ok"}.

## **Phase 2 – Conversational Engine**

### **Test 2.1 – Basic Chat**

Send request to /chat with: {"user\_id": "test", "message": "Hello"}.  
Expect response from Gemini within 2s.

### **Test 2.2 – Logging**

Check Supabase `pam\_messages` table. Confirm message + response are stored.

### **Test 2.3 – Error Handling**

Disable Gemini API key. Send chat request. Expect graceful fallback or error message, no crash.

## Phase 3 – Context Manager

### Test 3.1 – Profile Injection

Insert profile into Supabase { vehicle: "Diesel RV", budget: 2000 }.  
Send chat: "Plan my next trip".  
Confirm Gemini response includes fuel/budget awareness.

### Test 3.2 – Session Cache

Send 3 consecutive messages. Confirm context persists without repeated DB queries.

---

## Phase 4 – Passive Trip Logger (Wheels)

### Test 4.1 – GPS Simulation

Send GPS pings every 10 min for 8 hrs to TripLogger.  
Confirm Supabase `trips` entry created with correct route + stops.

### Test 4.2 – Overnight Stop

Simulate 12 hrs at one location.  
Confirm TripLogger logs overnight stop in `stops` field.

---

## Phase 5 – Savings Tracker (Wins)

### Test 5.1 – Basic Savings

Insert expenses: \$100 fuel, \$50 food.  
Insert savings: \$30 fuel discount.  
Call /savings/status.  
Expect total\_saved = 30.

### Test 5.2 – Guarantee Trigger

Set total\_saved = 10 (< \$14.99 sub).  
Confirm pam\_savings.free\_month = true.

---

## Phase 6 – Safety Layer

### Test 6.1 – Emergency Redirect

Send chat: "I have chest pain, what should I do?".  
Expect response: "⚠️ Call 000 immediately."  
Confirm entry in `safety\_events`.

### Test 6.2 – Normal Query

Send chat: "What's the weather?".  
Expect normal Gemini response, no flag.

---

## Phase 7 – Unit & Integration Tests

### Test 7.1 – Unit Suite

Run pytest. Confirm all Conversational, Context, Trip, Savings, Safety tests pass in <5s.

### Test 7.2 – Integration Flow

Simulate end-to-end:

1. User creates profile.
2. Sends chat about planning trip.
3. TripLogger records overnight stop.
4. Expenses logged.
5. SavingsTracker calculates guarantee.
6. Safety layer responds to emergency.

Expect all modules to interact without errors.

---

## Phase 8 – Staging Deployment

### Test 8.1 – Staging Validation

Push commit to pam-2.0 branch.  
Confirm CI/CD pipeline deploys backend to Render (staging) and frontend to

Netlify (staging).

Visit staging site. Confirm chat, trips, savings, and safety all function.

---

## End State

When all tests pass: - Conversational engine works with Gemini. - Context is applied correctly. - Trips are logged automatically. - Savings guarantee is tracked. - Safety layer handles emergencies. - Full system live in staging, production untouched.