MAHAKAL

Google Analytics 4 & Conversion Tracking

Implementation Documentation • Scope, Event Plan, Ads Integration & QA

Project: Mahakal website (Laravel / Blade frontend)

GA4 Measurement ID: G-BKWL2WND31

Document status: Planning and implementation guide based on the code checks shared so far

Purpose: establish reliable measurement across Mahakal's website modules, bookings, payments and marketing campaigns—not just basic page views.

Prepared: 27 October 2026

1. Executive Summary

Mahakal already has a basic GA4 installation using Measurement ID **G-BKWL2WND31**. The existing browser Network request to **google-analytics.com/g/collect** includes **en=page_view**, and the user has reported seeing activity in GA4 Realtime. This confirms that at least basic page-view collection is working in the checked flow.

The next stage is to audit every relevant module and implement consistent event tracking for important user actions, successful bookings/orders, payment outcomes and advertising conversions. Event names and parameters must reflect the real business flow and must be tested to avoid duplicate or misleading conversion counts.

Important: this document is a proposed scope and implementation plan. It does not claim that every event listed below is already implemented, nor that every listed module is present in the current production build. Confirm the module inventory and code paths during the audit before making changes.

2. Current Verified Status

Area	Status based on checks shared	Next action
GA4 base tag	Direct GA4 code found in default frontend Blade layout; Measurement ID G-BKWL2WND31.	Audit all active layouts/themes to ensure one intended page-view implementation per page.
Page view	A GA4 collect request with en=page_view was observed; Realtime shows activity.	Verify representative page types and prevent duplicate page_view events.
Google Tag Manager (GTM)	Dynamic google_tag_manager_id setting and conditional GTM snippets were found in code.	Confirm whether a GTM container ID (GTM-...) is configured and actually needed.
Custom module events	A targeted search did not find common gtag event/dataLayer patterns in app/resources.	Run a wider audit; the search result alone does not prove all custom events are absent.
Purchase/booking events	The reviewed COD checkout-completion controller creates orders and returns the completion view; no GA4 purchase event was seen in the reviewed function.	Audit all success/failure/payment flows and implement only after the correct source of truth is identified.
Google Ads	Account linking and Ads conversion configuration have not been confirmed.	Check Ads account, conversion actions, auto-tagging and GA4 linking.

3. Goals and Success Criteria

- Measure page traffic and engagement across all active Mahakal layouts and module entry points.
- Understand the user journey from landing page/search through lead, booking, checkout, payment and completion.
- Record successful commercial transactions once, with a stable transaction/booking ID and correct value/currency where applicable.
- Separate completed, failed, cancelled and pending actions so reports do not count unsuccessful transactions as revenue.
- Make GA4 reports and key events useful for business decisions and Google Ads conversion optimization.
- Document the event names, parameters, triggers, owners and test evidence so future developers can maintain tracking.

4. Phase 1 — Audit Before Coding

Do not begin by adding event snippets to multiple templates. First identify all active layouts, module controllers, checkout/booking routes, payment gateways, success pages and asynchronous/AJAX flows.

4.1 Existing analytics-code search (PowerShell)

Run this from the project root: **C:\xampp\htdocs\mahakal\mahakal**.

```
Get-ChildItem .\app,.\resources,.\config,.\public -Recurse -File -Include *.php,*.js,*.html,*.json
|
Where-Object { $_.FullName -notmatch '\\(vendor|node_modules|storage)\\' } |
```

```
Select-String -Pattern 'G-[A-Z0-9]+|GTM-[A-Z0-9]+|AW-[0-9]+|gtag\s*\(|dataLayer\.push|google_conversion_id|googleadservices\.com' |
Select-Object Path, LineNumber, @{Name='Code';Expression={$_.Line.Trim()}} |
Format-List
```

Review results for measurement IDs, container IDs, Ads conversion IDs, event calls and duplicate tags. Redact access tokens, API keys and other secrets before sharing command output.

4.2 Inventory commands

```
Get-ChildItem .\app\Http\Controllers -Directory | Select-Object Name
Get-ChildItem .\resources\themes -Directory | Select-Object Name
```

Then map each module to its actual route/controller, Blade page, booking/order table, payment callback, and final success state. Include alternate themes and mobile/API-driven flows where applicable.

5. Phase 2 — Baseline Traffic and Engagement

Measurement	Purpose / validation
page_view	Page loads and page paths; check each active layout and avoid duplicate firing.
session_start / engagement	Normally collected automatically by GA4 when the base tag is configured correctly.
scroll / outbound_click / file_download	Use Enhanced Measurement where suitable; test that the collected behavior matches expectations.
site_search	Track searches only when the site exposes a reliable search term parameter; do not send personal or sensitive values.
login / sign_up	Track successful authentication or registration, not merely a button click.
generate_lead	Use for a genuine lead/request submission, where relevant to a module.

6. Phase 3 — Module-wise Tracking Plan

The following is a candidate inventory based on Mahakal features discussed so far. Confirm whether each module is live, which user journey applies, and whether the event should be sent from the browser or server.

Module / journey	Candidate events	Success condition / notes
Pooja Shop / e-commerce	view_item, add_to_cart, remove_from_cart, view_cart, begin_checkout, purchase, refund	Use actual product IDs, item names, quantity, value and currency. Purchase only after verified successful order/payment.
Pooja booking	pooja_view, pooja_booking_start, pooja_booking_success, pooja_booking_failed, pooja_booking_cancelled	Use booking ID; distinguish paid, COD, pending and confirmed states.
Anushthan	anushthan_view, anushthan_booking_start, anushthan_booking_success, anushthan_payment_success	Define success according to the real business status and payment rules.
Astrologer / chat / call	astrologer_view, astrologer_profile_view, chat_request, call_request, consultation_start, consultation_complete	Do not count a click as a completed consultation; track accepted/connected/completed states separately.
Live stream / pooja stream	live_stream_view, live_stream_join, live_stream_end (where meaningful)	Avoid sending high-volume heartbeat events; define whether viewer engagement is a business KPI.
Tours / travel / cab	tour_view, generate_lead, tour_booking_start, tour_booking_success, pickup_otp_flow (only if useful)	Track booking and payment status; never send OTP values or personal details.
Parking	parking_search, parking_view, parking_booking_start, parking_booking_success, parking_payment_success	Confirm actual parking flow and stable reservation identifier.
Blog / Rashifal / Panchang	page_view, search, content_view (optional custom event)	Prefer page/content reports unless a custom event answers a specific business question.

Module / journey	Candidate events	Success condition / notes
Login / registration	login, sign_up, generate_lead (where applicable)	Send after server-confirmed success; never include email, phone or name in event parameters.
Other live modules	Module-specific view, start, success, failure and cancel events	Add only after inventory and KPI approval.

7. Event Naming and Parameter Standards

Use GA4 recommended event names when they fit the action. Use custom names only where the recommended taxonomy does not represent the business action. Keep names lowercase and consistent; document all custom events in an event dictionary.

Parameter	Usage
transaction_id	Stable unique order/booking ID; prevents duplicate transaction reporting when implemented correctly.
value	Numeric total value using the agreed business definition; do not send formatted strings.
currency	ISO 4217 currency code, e.g. INR.
items	For e-commerce, item_id/item_name, price and quantity as applicable.
payment_method	Only a non-sensitive method label if useful and approved; never send payment credentials.
module_name	A stable module label for custom events, e.g. pooja, astrologer, tour, parking.
booking_status	A controlled status label such as confirmed, pending, failed or cancelled.
method	For login/sign_up where applicable, use a non-sensitive method label.

Never send personally identifiable information (PII) such as names, email addresses, phone numbers, full addresses, OTPs, passwords, payment card details, or raw personal search text to GA4. Use internal IDs only where permitted and ensure they cannot directly identify a person.

8. Phase 4 — Booking and Transaction Tracking

This phase requires special care because a success page can be refreshed, revisited or opened after a redirect. A browser-only purchase event may be missed by ad blockers or repeated if the user reloads the page. Select a source of truth and implement duplicate protection.

Recommended flow

1. Identify the authoritative status transition that means an order or booking is truly successful (for example, payment captured or business booking confirmed).
2. Create an event mapping for each payment method: online success, COD, offline/manual payment, pending, failure, refund and cancellation.
3. Use a stable transaction_id or booking ID; do not generate a new ID each time the success page loads.
4. Fire the purchase/booking-success event only after the agreed success condition is met. Do not count pending or failed payments as purchases.
5. Add duplicate prevention and test refresh, back-button, repeated callback/webhook and retry scenarios.
6. For server-side delivery, evaluate GA4 Measurement Protocol only with correct consent handling, client/session attribution parameters where available, and secure server-side configuration.
7. Reconcile GA4 transaction counts and revenue against the application/database and payment gateway over a controlled test period.

Recommended GA4 e-commerce purchase fields (where applicable)

- transaction_id — stable unique order identifier

- value — total value defined consistently by the business
- currency — INR if the transaction is in Indian rupees
- tax and shipping — only if the system calculates them and they are relevant
- items — item identifiers, names, prices and quantities

For non-shop modules such as pooja, anushthan, tours, astrologer consultations or parking, decide whether the event is a GA4 purchase or a custom booking-success event based on the transaction model. Avoid counting the same transaction as both a purchase and a second revenue conversion in the same KPI without clear reporting rules.

9. Phase 5 — GA4 Key Events, Reports and Funnels

After events are validated in DebugView/Realtime, mark only business-critical events as key events (formerly conversions). Suggested candidates:

- purchase — confirmed revenue transaction, where applicable
- pooja_booking_success / anushthan_booking_success — confirmed bookings
- tour_booking_success / parking_booking_success — confirmed bookings, if those modules are active
- generate_lead — successfully submitted qualified lead/request
- sign_up — successful registration, if registration growth is a KPI

Build reports/funnels for: landing page → module view → start booking/checkout → payment attempt → success; and separately for failure/cancellation. Compare by source/medium/campaign, device category, landing page and module. Use Explorations or a dashboard for cross-module funnels where standard reports are insufficient.

10. Phase 6 — Google Tag Manager (GTM) Decision

The code audit found existing conditional GTM integration based on the `google_tag_manager_id` business setting, while GA4 is also hardcoded directly in a Blade layout. This can be valid, but only if tag ownership is clearly defined. Loading GA4 directly and again through GTM can create duplicate page_view or event counts.

Option	When to choose it	Control needed
Keep direct GA4	A small, stable setup with few tags and code-controlled releases.	Maintain one tag owner per page and test all layouts.
Use GTM as primary tag manager	Marketing team needs to manage GA4/Ads tags without each change requiring a code release.	Move/disable duplicate direct GA4 configuration carefully; publish and test a versioned GTM container.

Do not add a GTM container merely because the setting exists. First confirm whether a valid GTM-... ID is configured, whether the container is published, and whether it already contains a GA4 Google tag or Ads tags.

11. Phase 7 — Google Ads Integration

1. Confirm the correct Google Ads account and GA4 property permissions.
2. Link the GA4 property to Google Ads and enable the appropriate personalized advertising/data-sharing settings only when permitted by policy and consent.
3. Enable auto-tagging in Google Ads and verify campaign parameters/attribution on real test visits.
4. Choose the primary conversion source for each goal: import validated GA4 key events or configure Google Ads conversion tags. Avoid counting the same action twice as separate primary conversions.
5. Set conversion counting rules appropriately: transactions usually count every unique purchase; lead goals may use one or every conversion depending on business definition.
6. Validate conversion value/currency, attribution windows and campaign attribution before optimizing bids.
7. If remarketing, enhanced conversions or Consent Mode is planned, review applicable Google policies, consent requirements, implementation dependencies and regional obligations first.

Google Ads account linking and conversion tracking are not confirmed as completed in the current audit. Account setup, campaign creation and campaign optimization are separate workstreams and may extend the timeline.

12. Phase 8 — Privacy, Consent and Data Quality

- Review the privacy policy and cookie/consent behavior for the jurisdictions and audiences served.
- Configure consent behavior and Google Consent Mode where legally and operationally appropriate; do not assume a banner alone proves correct consent signaling.
- Exclude internal/test traffic from production reporting where feasible, and document how it is identified.
- Check unwanted referrals/payment-gateway referrals so attribution is not broken, without hiding genuine acquisition sources.
- Review cross-domain measurement only if a user journey legitimately moves between domains and the setup requires continuity.
- Use data retention and access controls appropriate for the business; limit access to GA4 and Ads accounts.
- Do not place personal data in URLs, page titles, event names, search terms or event parameters.

13. Phase 9 — Testing and Go-Live Checklist

Test	Pass condition
Base tag / page_view	One intended page_view per page load in each supported layout; correct Measurement ID.
Module navigation	Representative module pages report expected page path and source/medium.
Custom events	Correct event name and parameters visible in DebugView/Realtime.
Successful booking	Exactly one success event after the authoritative success condition.
Failed/pending/cancelled booking	Not incorrectly counted as a purchase; status reporting is accurate.
Duplicate protection	Refresh, back navigation, repeated callback and retry do not create duplicate transactions.
Revenue reconciliation	Test transaction IDs, values and currencies match application/payment records.
Ads conversion	One intended primary conversion source; test and attribution checks completed.
Privacy	No PII or secrets in GA4 payloads, URLs or logs; consent behavior reviewed.
Regression	Checkout, login, booking and payment flows still work after release.

Use browser Developer Tools Network, GA4 DebugView and Realtime for testing. Realtime is useful for immediate confirmation but should not be treated as the sole validation of attribution or long-term reporting. Document test date, route, test transaction ID, expected event, observed event and result.

14. Deliverables

- Existing tracking audit and active-layout inventory.
- Module inventory with controller/routes, completion conditions and database/payment source of truth.
- Approved event dictionary (event name, trigger, parameters, module, owner, key-event status).
- Implemented and tested event tracking for agreed modules and transactions.
- GA4 key-event configuration and useful baseline reports/funnels.
- Google Ads link/conversion setup if the Ads account and access are available.
- QA evidence and a release/rollback checklist.
- Handover notes explaining where future events should be maintained.

15. Indicative Timeline

Workstream	Estimated effort
Audit current tags, layouts and IDs	0.5–1 working day
Confirm modules and booking/payment flows	0.5–1.5 working days
Finalize event dictionary	0.5–1 working day
Implement module-level events	1–3 working days
Booking/purchase integration and deduplication	1–2 working days
GA4 key events, reports and Ads linking	0.5–1 working day
QA, reconciliation and fixes	1–2 working days
Overall planning estimate	Approximately 5–10 working days for one developer working regularly

This is an initial estimate, not a commitment. The final estimate depends on the number of active modules, how many payment gateways and success flows exist, whether the current code has consistent booking statuses, the need for server-side events, and whether Google Ads/GTM accounts are already configured. Ads campaign setup and ongoing optimization are not included.

16. Recommended Implementation Order

1. Complete the code/layout/ID audit and list all active modules.
2. Map each module's success condition to its controller, database status and payment callback.
3. Approve the event dictionary and define which events are key events.
4. Implement baseline and module events in a consistent location or GTM strategy.
5. Implement booking/purchase events with transaction IDs and duplicate prevention.
6. Configure GA4 key events, reports/funnels and Google Ads integration if access is ready.
7. Test in a controlled environment, reconcile transactions, then release to production.
8. Monitor after release and document any expected differences between GA4 and internal reports.

Appendix A — Important Existing Code Locations

Location	Known finding from checks shared
resources/themes/default/layouts/front-end/app.blade.php	Direct GA4 snippet with G-BKWL2WND31; conditional GTM loading also present in the file.
resources/themes/default/layouts/temple-front-end/app1.blade.php	Direct GA4 snippet found; verify whether this layout is active and whether other layouts also load GA4.
resources/views/admin-views/business-settings/analytics/index.blade.php	Admin setting UI reads google_tag_manager_id.
app/Http/Controllers/Admin/Settings/BusinessSettingsController.php	Update/insert logic for the google_tag_manager_id setting.
app/Http/Controllers/Web/WebController.php — checkout_complete()	Reviewed COD order-completion flow creates orders and returns completion view; no GA4 purchase event was seen in that reviewed function.
resources/themes/default/web-views/checkout/complete.blade.php	Order completion template identified; inspect how it receives order IDs and whether it is shared by multiple payment flows.

Appendix B — Definition of Done

- Every active layout and module has been audited.
- The approved event dictionary matches real application behavior.
- No duplicate base tag or duplicate transaction events occur in tested journeys.
- Successful, failed, pending and cancelled outcomes are represented correctly.
- Transaction IDs and revenue parameters are accurate and reconciled.
- GA4 key events and Ads conversion sources are configured without double-counting.
- Privacy/consent review and production QA are complete.
- Implementation and maintenance notes have been handed over.

End of document • This guide is a scope and implementation plan. Validate all assumptions against the live code, account configuration, business rules and applicable privacy requirements before deployment.