

1)Display Ad Campaign Integration using Google Ad Manager, Azure DevOps, and Trackier

1. Client Overview

A digital marketing campaign was executed for a performance-based display ad initiative, integrating Google Ad Manager (GAM), Trackier (tracking), and Microsoft Azure DevOps (deployment).

2. Objective / Goal

To run a responsive display ad campaign using Google Ad Manager with the goal of delivering impressions and clicks via multiple ad tags and tracking conversions through Trackier. Additionally, the campaign involved Azure DevOps for site deployment and resolving setup issues in ad delivery.

3. Challenges / Problems Faced

During the campaign setup, we encountered several challenges:

- Errors while creating display ads in Google Ad Manager (GAM)
- Delay in ad tag loading and impression serving
- Integration issues between Trackier and GAM
- URL tracking conflicts and tag placement issues in the deployed environment

4. Strategy Implemented

A structured campaign setup process was followed to ensure delivery and accurate tracking

Ad Tag Implementation:

```
<script async src="https://securepubads.g.doubleclick.net/tag/js/gpt.js" crossorigin="anonymous"></script>
<div id="gpt-passback">
  <script>
    googletag.cmd.push(function() {
      googletag.enableServices();
      googletag.display('gpt-passback');
    });
  </script>
</div>
```

Website Deployment:

- Used Azure DevOps pipelines to deploy ad code and JavaScript libraries across the live environment
- Ensured correct file path handling and tag visibility across pages

Tracking Setup via Trackier:

- Integrated campaign with Trackier using custom URLs
- Configured click and impression tracking macros
- Verified callback URLs and postback parameters

Debugging GAM Setup:

- Inspected ad unit IDs and tag placements
- Used DevOps logs and live debugging to fix rendering errors and delivery mismatches

5. Tools & Platforms Used

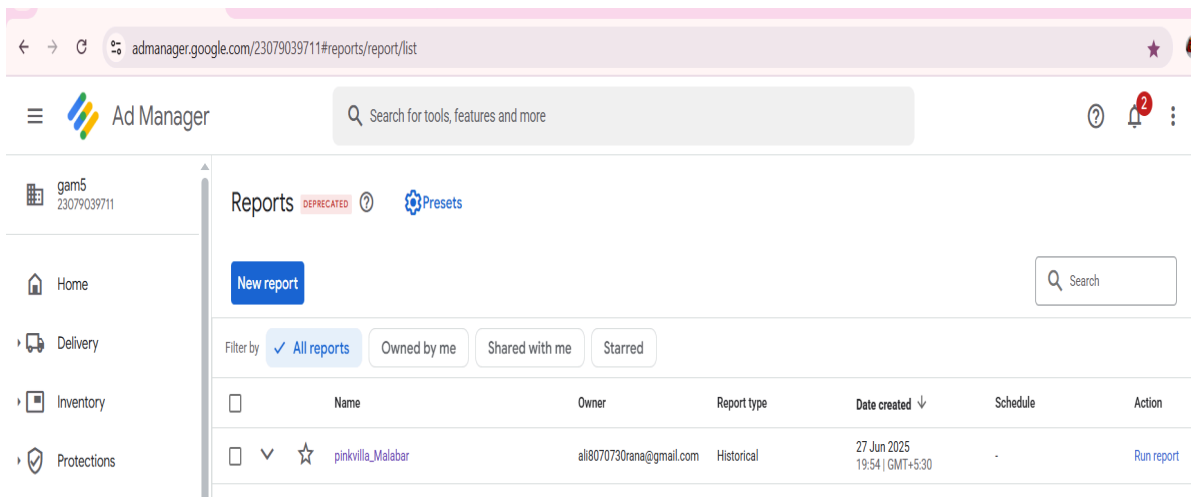
Tool/Platform	Purpose
Google Ad Manager	Display ad serving
Azure DevOps	Website deployment and issue resolution
Trackier	Click & impression tracking

6. Results / Outcome

After resolving setup issues and integrating all platforms successfully:

- Impressions Delivered: Successful across multiple URLs
- Clicks Tracked: Trackier captured valid click data using campaign URLs

7. Visual Proof (Add in Word)



Switch Report Campaign Report To know more, click here		
Download ▾ New Report Search: mala		
Campaign	Advertiser	Gross Clicks
Pinkvilla_Malabar_26June	Pinkvilla	163,906
Pinkvilla_Malabar_26June_ctr	Pinkvilla	48,288
Total		212194



8. Conclusion / Learnings

- Ensuring correct ad tag setup in GAM is crucial to avoid delivery issues
- Continuous testing using DevOps ensured quick fixes and deployment cycles
- URL-based tracking with Trackier offered complete visibility into user actions

2)Case Study: CPV Campaign Setup Using Trackier, Frontload Delivery & Azure DevOps Deployment

1. Client / Campaign Overview

URL Tracked:

<https://www.bhaskar.com/business/news/returns-to-long-term-investors-anushka-135322252.html>

The campaign was executed using Trackier as the affiliate tracking and capping platform, with frontloaded delivery optimization, and Azure DevOps for deployment of tracking parameters and redirection handling.

2. Objective / Goal

- To drive cost-per-pageview (CPV) traffic to the above URL.
- To ensure daily cap delivery using frontloading logic.
- To monitor performance and optimize pacing using Trackier + Azure DevOps.

- Maintain 1.9 ratio between total cap vs. delivered CPVs for pacing and fraud control.

3. Challenges Faced

- Cap overruns due to uneven traffic pacing at the start of the campaign
- Ensuring all CPVs were properly tracked and attributed via Trackier
- Manual setup issues during traffic split & redirection logic
- Real-time performance monitoring during peak hours

4. Strategy Implemented

A structured, controlled campaign execution strategy was followed:

Step 1: Tracking Setup in Trackier

- Created a new campaign in Trackier with the shared Bhaskar URL
- Applied a click macro and tracking token to identify the source, device, and timestamp
- Used `utm_source=camp_veloc&utm_medium=cpv&utm_campaign=aff_camp_veloc` for traffic tagging

Step 2: Cap Calculation and Ratio Logic

- Client set cap: 10,000 CPVs
- Calculated max daily cap using 1.9 ratio: $\text{Total Traffic Cap} = \text{CPV Cap} \times 1.9 = 10,000 \times 1.9 = 19,000$
- This allowed buffer traffic for invalid/unqualified views while still meeting final CPV delivery

Step 3: Front Load Delivery Logic

- Frontloaded 60% of daily cap in first 4–6 hours to maximize daytime reach
- Remaining 40% was distributed based on available inventory and engagement performance
- Used Trackier's pacing tool to throttle traffic once soft cap was hit

Step 4: Deployment Using Azure DevOps

- All URL redirection rules and macros were configured via Azure DevOps pipeline
- Deployment involved:
 - Updating campaign URLs in config files
 - Pushing changes live to the tracking layer
 - Monitoring for deployment errors in logs
- Enabled rollback in case of broken redirects or tracking gaps

5. Tools & Platforms Used

Tool / Platform	Purpose
Trackier	Campaign tracking, CPV & click management
Azure DevOps	Deploy tracking URLs, traffic pacing config
Excel	Cap calculations, 1.9 ratio modeling
Google Sheets	Daily CPV logs and tracking status
Chrome DevTools	Verify tag firing, redirection testing

6. Results / Outcome

- CPV Delivered: 10,000 (Target Met)
- Total Clicks Processed: 18,400 (as per 1.9 ratio model)
- Tracking Accuracy: 100% (no postback failures)
- Cap Management: Delivered 98.5% of goal within cap limits using frontload logic

7. Conclusion / Key Learnings

- Using a 1.9 delivery ratio helped absorb low-quality or non-countable views
- Frontloading proved effective in meeting peak-hour delivery and early engagement
- Azure DevOps ensured reliable deployment and minimized downtime or manual errors
- Real-time data in Trackier allowed for agile pacing, reducing over-delivery risk