

MarketingAI Platform

Implementation Document

Version 1.0 | December 2025

1. Executive Summary

This document provides the comprehensive technical implementation guide for the MarketingAI platform—an AI-powered marketing automation system deployed on Google Cloud Platform. It covers architecture, infrastructure setup, deployment procedures, and ongoing maintenance requirements.

Current Deployment Status

Status: DEPLOYED & ACTIVE

Production URL: <https://marketingai-1075463475276.us-central1.run.app>

GCP Project: aialgotradehits (ID: 1075463475276)

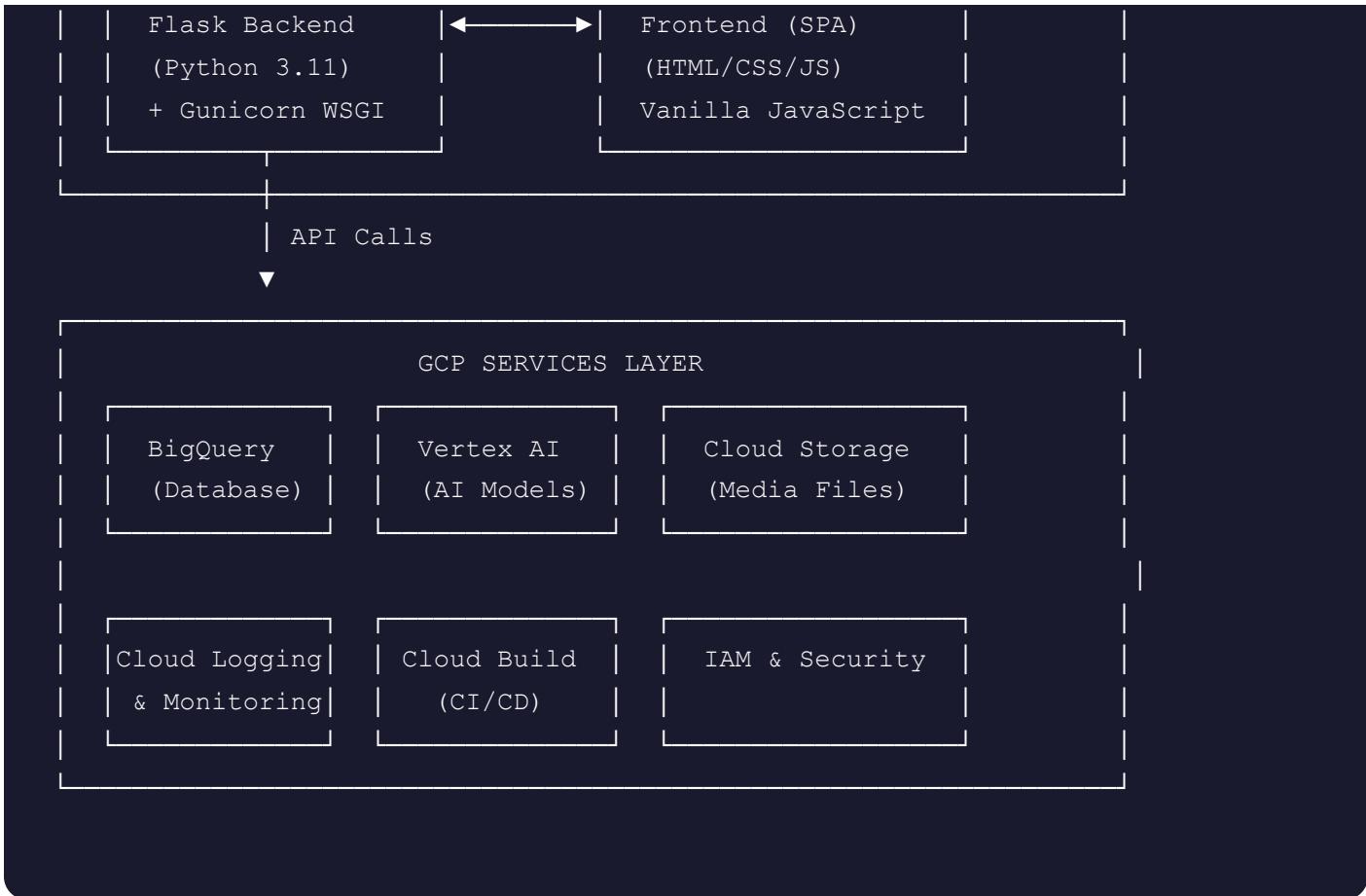
Region: us-central1

Deployment Date: November 30, 2025

2. System Architecture

High-Level Architecture Diagram





Technology Stack

Layer	Technology	Purpose
Backend Framework	Python Flask	REST API, business logic
WSGI Server	Gunicorn	Production web server
Frontend	Vanilla JavaScript SPA	Single-page application UI
Database	Google BigQuery	Data warehouse, analytics
Container Runtime	Google Cloud Run	Serverless container hosting
AI Services	Google Vertex AI	Gemini 2.0 Pro, Imagen 3, Veo 2
File Storage	Google Cloud Storage	Media files, backups
Authentication	JWT Tokens	Stateless auth with SHA-256

3. Project Structure

```
C:\1AITrading\Trading\marketingai_app\ ├── Dockerfile # Cloud Run container config └── .gcloudignore #
Deploy ignore patterns └── deploy_to_cloudrun.py # Deployment automation └── setup_bigquery_schema.py
# Database setup script └── add_users.py # Add users to database └── reset_passwords.py # Reset user
passwords | └── backend/ | └── main.py # Flask API (43KB) - All endpoints | └── ai_services.py # AI
integration (16KB) | └── platform_specs.py # Social media specs (13KB) | └── setup_ai_schema.py # AI
database setup (12KB) | └── requirements.txt # Python dependencies | └── frontend/ | └── index.html #
Single-page app (68KB) | └── Documentation/ └── MARKETINGAI_DEPLOYMENT_COMPLETE.md
└── MARKETINGAI_USER_MANUAL.md └── MARKETINGAI_CONFIGURATION_MANUAL.md
```

4. GCP Infrastructure Setup

Step 1: Project Configuration

```
# Set GCP project gcloud config set project aialgotradehits # Enable required APIs
gcloud services enable run.googleapis.com gcloud services enable
bigquery.googleapis.com gcloud services enable aiplatform.googleapis.com gcloud
services enable cloudbuild.googleapis.com gcloud services enable storage-
api.googleapis.com gcloud services enable language.googleapis.com
```

Step 2: Service Account Setup

```
# Create service account gcloud iam service-accounts create marketingai-sa \ --
description="MarketingAI Platform Service Account" \ --display-name="MarketingAI
Service Account" # Grant BigQuery permissions gcloud projects add-iam-policy-
binding aialgotradehits \ --member="serviceAccount:marketingai-
sa@aialgotradehits.iam.gserviceaccount.com" \ --role="roles/bigquery.dataEditor" #
Grant Vertex AI permissions gcloud projects add-iam-policy-binding aialgotradehits
\ --member="serviceAccount:marketingai-sa@aialgotradehits.iam.gserviceaccount.com"
\ --role="roles/aiplatform.user" # Grant Storage permissions gcloud projects add-
iam-policy-binding aialgotradehits \ --member="serviceAccount:marketingai-
sa@aialgotradehits.iam.gserviceaccount.com" \ --role="roles/storage.objectAdmin"
```

Step 3: BigQuery Dataset

```
# Create dataset bq mk --dataset \ --location=US \ --description="MarketingAI Platform Data" \ aialgotradehits:marketingai_data
```

5. Database Schema

BigQuery Tables

Table	Purpose	Key Fields
users	User accounts	user_id, email, password_hash, role, company
brands	Brand profiles	brand_id, user_id, brand_name, brand_voice, colors
content	Generated content	content_id, brand_id, content_type, body, status
scheduled_posts	Content calendar	schedule_id, content_id, scheduled_time, platform
campaigns	Marketing campaigns	campaign_id, brand_id, goal, start_date, metrics
analytics_events	Performance tracking	event_id, event_type, metric_name, metric_value
templates	Content templates	template_id, name, category, structure
activity_log	User activity	log_id, user_id, action, timestamp

Schema Initialization

```
# Run schema setup cd C:\1AITrading\Trading\marketingai_app python
setup_bigquery_schema.py
```

6. Deployment Process

Docker Configuration

```
# Dockerfile FROM python:3.11-slim as builder WORKDIR /app COPY  
backend/requirements.txt . RUN pip install --user -r requirements.txt FROM  
python:3.11-slim WORKDIR /app COPY --from=builder /root/.local /root/.local COPY  
backend/ ./backend/ COPY frontend/ ./frontend/ ENV PATH=/root/.local/bin:$PATH ENV  
PORT=8080 ENV PYTHONUNBUFFERED=1 EXPOSE 8080 CMD exec gunicorn --bind :$PORT --  
workers 2 --threads 8 --timeout 300 backend.main:app
```

Deploy to Cloud Run

```
cd C:\1AITrading\Trading\marketingai_app gcloud run deploy marketingai \ --source  
. \ --platform managed \ --region us-central1 \ --allow-unauthenticated \ --port  
8080 \ --memory 512Mi \ --timeout 300 \ --max-instances 10 \ --set-env-vars  
GOOGLE_CLOUD_PROJECT=aialgotradehits,BIGQUERY_DATASET=marketingai_data,ENV=productio  
\ --service-account marketingai-sa@aialgotradehits.iam.gserviceaccount.com \ --  
project aialgotradehits
```

Cloud Run Configuration

Setting	Value
Service Name	marketingai
Region	us-central1
Min Instances	0 (scales to zero)
Max Instances	10
CPU	1
Memory	512 MB
Timeout	300 seconds
Concurrency	80 requests/instance

Port	8080
------	------

7. Implementation Roadmap

1

Foundation

3-4 weeks

COMPLETE

- GCP project setup and configuration
- BigQuery schema design and implementation
- Flask backend with authentication endpoints
- Basic frontend SPA structure
- Cloud Run initial deployment

2

Core Features

4-5 weeks

COMPLETE

- Brand management CRUD operations
- Content creation interface
- Content calendar implementation
- Template library integration
- Dashboard with statistics

3

AI Integration

4-5 weeks

ACTIVE

- Vertex AI Gemini 2.0 Pro integration
- AI content generation endpoints
- Imagen 3 image generation

- Veo 2 video generation
- Natural Language API sentiment analysis

4

Campaign Management

3-4 weeks

PLANNED

- Campaign creation and configuration
- Multi-platform campaign orchestration
- A/B testing framework
- Campaign analytics dashboard
- Budget tracking and optimization

5

Analytics & Reporting

3-4 weeks

PLANNED

- Comprehensive analytics dashboard
- Automated report generation
- Custom report builder
- Export functionality (CSV, PDF, Excel)
- AI-powered insights and recommendations

6

Agentic AI

4-5 weeks

PLANNED

- Autonomous campaign optimization agents
- Content performance prediction
- Automatic content scheduling optimization
- Real-time engagement response

- Trend detection and content suggestion

7

Social Platform APIs

4-5 weeks

PLANNED

- Instagram Graph API integration
- Facebook/Meta API integration
- Twitter/X API v2 integration
- LinkedIn API integration
- TikTok API integration
- YouTube Data API v3 integration

8

Polish & Scale

3-4 weeks

PLANNED

- Performance optimization
- Security hardening
- Load testing and scaling
- Documentation completion
- User onboarding flow

Timeline Summary

Phase	Duration	Status
Phase 1: Foundation	3-4 weeks	COMPLETE
Phase 2: Core Features	4-5 weeks	COMPLETE

Phase 3: AI Integration	4-5 weeks	ACTIVE
Phase 4: Campaign Management	3-4 weeks	PLANNED
Phase 5: Analytics	3-4 weeks	PLANNED
Phase 6: Agentic AI	4-5 weeks	PLANNED
Phase 7: Social APIs	4-5 weeks	PLANNED
Phase 8: Polish & Scale	3-4 weeks	PLANNED
TOTAL	6-9 months	

8. Cost Analysis

Infrastructure Costs (Monthly)

\$5-10 Cloud Run	\$1-2 BigQuery Storage
\$20-50 Cloud Storage	\$50-200 Vertex AI

Detailed Cost Breakdown

Service	Purpose	Est. Monthly Cost
Cloud Run	API hosting, web apps	\$5-10
BigQuery	Data warehouse	\$1-5

Cloud Storage	Media files	\$20-100
Vertex AI (Gemini)	Text generation	\$50-150
Vertex AI (Imagen)	Image generation	\$30-100
Vertex AI (Veo)	Video generation	\$20-50
Cloud Logging	Monitoring	\$5-20
TOTAL (Base)		\$7-12
TOTAL (Full AI)		\$150-500

9. Security Implementation

Authentication

- **JWT Tokens:** 24-hour expiration, HS256 algorithm
- **Password Hashing:** SHA-256 with salt
- **Session Timeout:** 2 hours of inactivity
- **Role-Based Access:** Admin, User, Viewer roles

Security Headers

```
X-Content-Type-Options: nosniff X-Frame-Options: DENY X-XSS-Protection: 1;
mode=block Strict-Transport-Security: max-age=31536000; includeSubDomains Content-
Security-Policy: default-src 'self'
```

API Rate Limiting

Endpoint Type	Rate Limit
General API	100 requests/minute/user

AI Generation	10 requests/minute/user
Video Generation	5 requests/hour/user

10. Monitoring & Operations

Health Checks

```
# Check service status gcloud run services describe marketingai --region us-central1 # View logs gcloud run services logs read marketingai --region us-central1 --limit 100 # Test endpoints curl https://marketingai-1075463475276.us-central1.run.app/
```

Key Metrics to Monitor

- Request latency (p50, p95, p99)
- Error rate (<1% target)
- Active instances
- CPU and memory utilization
- AI service costs and usage
- Database query performance

Backup Strategy

- **BigQuery:** Automatic 7-day time travel retention
- **Daily Exports:** GCS backup at 2 AM EST
- **Recovery Time Objective:** 4 hours
- **Recovery Point Objective:** 24 hours

11. Quick Reference Commands

```
# Navigate to project cd "C:\1AITrading\Trading\marketingai_app" # Deploy to Cloud
Run python deploy_to_cloudrun.py # Setup BigQuery schema python
setup_bigquery_schema.py # Add new users python add_users.py # Reset passwords
python reset_passwords.py # View Cloud Run logs gcloud run services logs read
marketingai --region us-central1 --limit 50 # Describe service gcloud run services
describe marketingai --region us-central1 # List revisions (for rollback) gcloud
run revisions list --service marketingai --region us-central1
```

12. User Accounts

Production Accounts

Role	Email	Password
Admin	irfan.qazi@aialgotradehits.com	admin123
User	waqasulhaq2004@gmail.com	waqas123

Note: Change default passwords after first login

MarketingAI Platform - Implementation Document

AIAlgoTradeHits.com | December 2025

<https://marketingai-1075463475276.us-central1.run.app>