

GOFAST JOBS ZW

Master Database v2.0

Developer Ready Database Structure

This document defines the core database tables, fields, relationships, statuses, and API interaction logic required to build Gofast Jobs ZW, powered by Uncle Tonde.

1. Core Tables

users
workers
employers
jobs
applications
payments
verification_records
job_offers
notifications
broadcasts
admin_users
audit_logs
support_tickets

2. users Table

Purpose: Central login and user identity table.

Fields:

user_id	PK
user_type	worker / employer / diaspora_employer / admin
phone_number	unique
email	nullable
password_hash	nullable
whatsapp_id	nullable
account_status	active / suspended / deleted
created_at	
updated_at	
last_seen_at	

3. workers Table

Purpose: Stores complete worker profile.

Fields:

worker_id	PK
user_id	FK → users.user_id
full_name	
phone_number	
age	
gender	
city	

suburb
 province
 main_category Domestic Worker / Shop Assistant / Artisan-Tradesman
 sub_category Maid / Gardener / Retail Worker / Electrician etc.
 work_type Stay-In / Stay-Out / Part-Time / Full-Time / Per Job /
 Contract / Other
 experience_level Less than 1 year / 1-3 years / 3-5 years / More than 5
 years
 profile_photo_url
 availability_status available / interviewing / hired / inactive / suspended
 applications_used_this_month
 monthly_application_limit default 3
 next_of_kin_name
 next_of_kin_relationship
 next_of_kin_phone
 created_at
 updated_at

Rule:

Workers are free during launch but limited to 3 applications per month.

4. employers Table

Purpose: Stores local and diaspora employers.

Fields:

employer_id PK
 user_id FK → users.user_id
 full_name
 phone_number
 email
 location
 country
 diaspora_flag true / false
 employer_type individual / company / diaspora_family
 created_at
 updated_at

5. jobs Table

Purpose: Stores job adverts.

Fields:

job_id PK
 employer_id FK → employers.employer_id
 category Domestic Worker / Shop Assistant / Artisan-Tradesman
 sub_category
 work_type
 job_title
 job_location
 city
 suburb
 salary_amount
 salary_type monthly / weekly / daily / per-job / contract
 job_description
 workers_required

positions_filled
job_status draft / payment_pending / pending_approval / active /
closed / expired / rejected
created_at
approved_at
expires_at created_at + 30 days
closed_at

Rule:

Employer job posts cost \$2 and last 30 days.

6. applications Table

Purpose: Tracks worker applications.

Fields:

application_id PK
job_id FK → jobs.job_id
worker_id FK → workers.worker_id
application_status pending / shortlisted / accepted / rejected / withdrawn
applied_at
employer_response_at
worker_response_at

Rules:

If worker applications_used_this_month >= 3, block application.

Otherwise create application and increment applications_used_this_month.

7. job_offers Table

Purpose: Manages direct offers and hire confirmation.

Fields:

offer_id PK
job_id FK → jobs.job_id
worker_id FK → workers.worker_id
employer_id FK → employers.employer_id
offer_status sent / accepted / declined / failed / confirmed
sent_at
accepted_at
confirmed_at
failed_at
failure_reason

Rules:

If offer_status = accepted:

worker.availability_status = hired

job.job_status = closed

If offer_status = failed:

worker.availability_status = available

job.job_status = active

8. payments Table

Purpose: Stores EcoCash and Visa/Mastercard payments.

Fields:

payment_id	PK
user_id	FK → users.user_id
employer_id	nullable FK
job_id	nullable FK
payment_method	ecocash / visa / mastercard
payment_provider	ecocash / stripe / paynow / other
amount	
currency	USD / ZWL
payment_reference	
gateway_transaction_id	
gateway_response_json	
payment_status	pending / successful / failed / refunded / expired
created_at	
confirmed_at	

Rule:

If payment_status = successful, update linked job to pending_approval.

9. verification_records Table

Purpose: Tracks ID and police-certified verification documents.

Fields:

verification_id	PK
worker_id	FK → workers.worker_id
verification_type	national_id / police_certified_copy
document_url	
verification_status	pending / verified / rejected / expired
reviewed_by	FK → admin_users.admin_id
review_notes	
submitted_at	
reviewed_at	

10. notifications Table

Purpose: Message queue for WhatsApp alerts.

Fields:

notification_id	PK
user_id	FK → users.user_id
notification_type	job_alert / application_update / payment_update / verification_update / system
message_body	
delivery_channel	whatsapp / email / sms
delivery_status	pending / sent / failed
created_at	
sent_at	

11. broadcasts Table

Purpose: Admin mass communication.

Fields:

broadcast_id	PK
created_by	FK → admin_users.admin_id
broadcast_type	job_alert / worker_recruitment / employer_marketing /

diaspora_campaign / safety_awareness
audience workers / employers / diaspora / all
message_body
status draft / scheduled / sent / failed
scheduled_at
sent_at
created_at

12. admin_users Table

admin_id PK
full_name
email
phone_number
role super_admin / verification_admin / payments_admin /
support_admin / jobs_admin
status active / suspended
created_at
last_login_at

13. audit_logs Table

Purpose: Security and accountability.

Fields:

log_id PK
admin_id nullable FK
user_id nullable FK
action_type create / update / approve / reject / suspend / login /
payment_confirm
entity_type worker / employer / job / payment / verification
entity_id
old_value_json
new_value_json
ip_address
created_at

14. support_tickets Table

ticket_id PK
user_id FK → users.user_id
ticket_type payment / verification / job_issue / technical /
abuse_report
subject
message
status open / in_progress / resolved / closed
assigned_admin_id nullable FK
created_at
resolved_at

15. Key Relationships

users → workers
users → employers
employers → jobs
jobs → applications
workers → applications

jobs → job_offers
workers → job_offers
users → payments
jobs → payments
workers → verification_records
admin_users → audit_logs
admin_users → broadcasts
users → support_tickets

16. API Interaction Logic

Worker Registration API:

POST /api/workers/register

Creates users record, workers record, and optional verification_records record.

Job Posting API:

POST /api/jobs/create

Creates jobs record with status = payment_pending.

Payment Callback API:

POST /api/payments/callback

Updates payments.payment_status = successful and jobs.job_status = pending_approval.

Application API:

POST /api/applications/create

Checks monthly_application_limit, worker availability, and job active status.

Hire Confirmation API:

POST /api/offers/confirm

Updates job_offers.offer_status, workers.availability_status, and jobs.job_status.

17. Critical Developer Recommendation

Do not combine workers and employers into one overloaded table. Use a central users table, then separate workers and employers profile tables.

Benefits:

- Cleaner permissions
- Better future expansion
- Easier analytics
- Safer admin controls
- Better scalability