GOLANG
BACKEND
DEVELEPMENT
MASTERY
BATCH 1

Interactive CARES

Rule the Backend World with Go

Scalable সিস্টেম, APIs ও
Microservices বানিয়ে হ্যান্ডস-অন প্র্যাকটিসে
হয়ে উঠুন Golang Backend Developer!!!

- ৪০টি লাইভ ক্লাস
- ফিডব্যাক সেশন
- হ্যান্ডস অন প্রজেক্টস

GOLANG BACKEND DEVELEPMENT MASTERY
BATCH 1

Interactive CARES

Meet The Instructor

S M SHAHRIAR ISLAM

Department Manager,
Devops at CP Axtra

Join the journey to build the future of high-performance backends!!

GOLANG BACKEND DEVELEPMENT MASTERY
BATCH 1

Interactive CARES

কেন Golang Backend Development শিখবেন?

৬৫%
টাইম সেভ – concurrency ও efficient libraries দিয়ে টাস্ক ফাস্ট এক্সিকিউশন

৮০%
হাই-ডিমান্ড – Microservices, gRPC, Pub/Sub, Temporal workflow ডেভেলপারদের জন্য

80%
অপারেশনাল খরচ কমানো – scalable ও high-performance সিস্টেম তৈরি করে

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

কারা হতে পারেন Golang Backend Developer?

- Backend Engineer যারা স্কিল আপগ্রেড করতে চান
- Golang Developer যারা distributed systems এ কাজ করতে চান
- Software Engineer যারা concurrency ও gRPC শিখতে চান
- Tech Professional যারা system design ও microservices স্কিল বাড়াতে চান

GOLANG
BACKEND
DEVELEPMENT
MASTERY
BATCH 1

Interactive
CARES

প্রোডাকশন-গ্রেড Backend তৈরি করুন Golang দিয়ে - এনরোল করুন আজই!

আপনি যেকোনো প্রোগ্রামিং ব্যাকগ্রাউন্ড থেকে আসুন, এই কোর্সে শিখবেন
production-ready backend তৈরি করার স্কিল।
আপনার ব্যাকএন্ড স্কিলকে নেক্সট লেভেলে নিয়ে যান এখনই।

যেকোনো সমস্যায় পড়লে

কল করুন

আমাদের কাস্টমার কেয়ার নাম্বারে

+880 1958622164

শেখা শুরু করি

কমিউনিটিতে জয়েন করি

GOLANG BACKEND DEVELEPMENT MASTERY
BATCH 1

Interactive CARES

Golang Backend Development - Production-Ready Skills!

এই কোর্সে ধাপে ধাপে শিখবেন কীভাবে ব্যাকএন্ড সার্ভিস
ডিজাইন, ডেভেলপ ও ডেপ্লয় করবেন প্রোডাকশন-গ্রেড অ্যাপ্লিকেশন!

আপনি যা শিখবেন

- Golang fundamentals & concurrency
- REST & gRPC microservices
- Temporal workflows & Pub/Sub pipelines
- System design, observability & autoscaling
- CI/CD, Docker, Kubernetes deployment

Capstone Project: Ride-Sharing Backend - complete services + Temporal workflows + scaling + observability

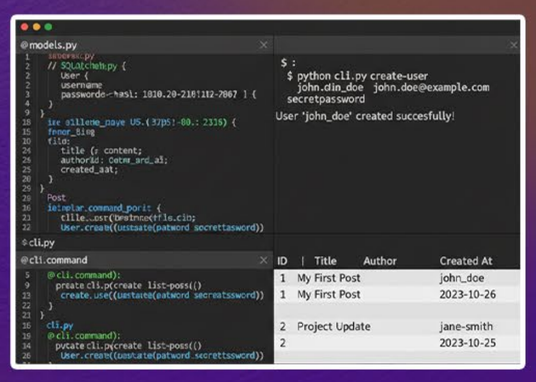
এক কোর্সে সব কিছু: Theory + Tools + Projects + Capstone Deployment

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

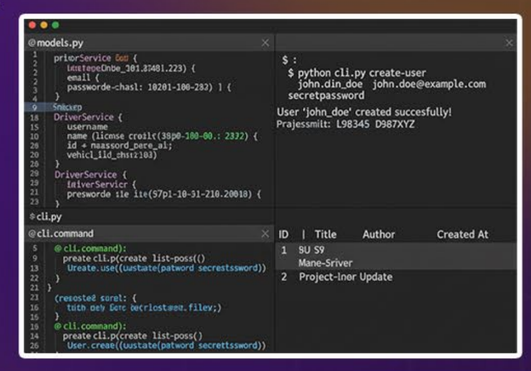
Interactive CARES

এই কোর্সে যে মিনি প্রজেক্টগুলো আপনি শিখবেন?

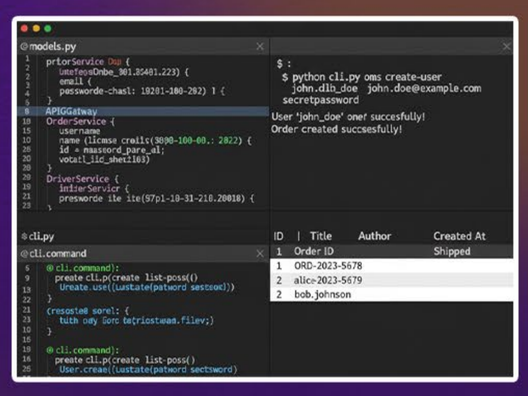
CLI Tools



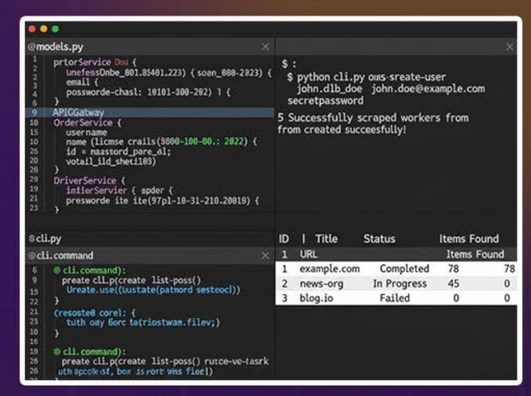
User Service & Driver Service



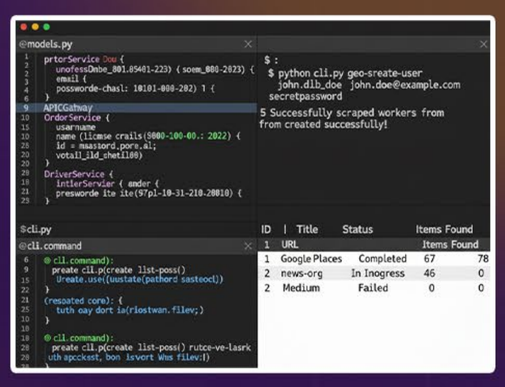
OMS + API Gateway



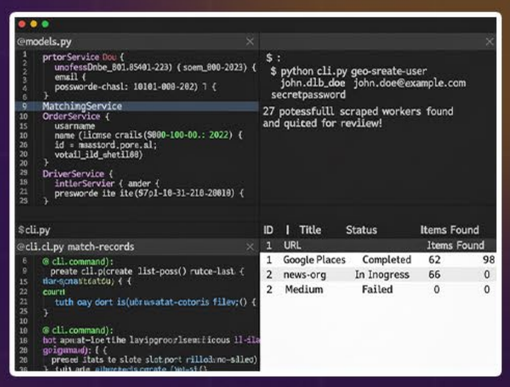
Concurrent Web Scraper



Geo Service



Matching Service



GOLANG BACKEND DEVELEPMENT MASTERY
BATCH 1

Interactive CARES

Core Tools & Tech:

Golang

Docker

Kubernetes

Redis

PostgreSQL

Elasticsearch

Kafka/
Redpanda

Temporal

gRPC

REST API

k6

Quickwit

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

FAQ (Frequently Asked Question)

✦ কোর্সের ক্লাস কবে থেকে শুরু?

- কোর্সের ক্লাস শুরু হবে জানুয়ারি থেকে।

✦ কোর্সের কোনো ফ্রি ডেমো ভিডিও আছে?

- হ্যাঁ, আমাদের ইন্সট্রাকটরদের একটি ফ্রি ডেমো ভিডিও দেখতে **এখানে ক্লিক করুন**।

✦ কোর্সে কী কী বার ক্লাস হবে?

- পরে জানিয়ে দেওয়া হবে।

✦ কোর্সে এনরোল করতে কত টাকা লাগবে?

- ৮০০০ টাকা দিয়ে এনরোল করে পেয়ে যান ক্যারিয়ার বিল্ড আপের জন্যে দেশসেরা গাইডলাইন।

✦ কোর্সটি কীভাবে কিনবো?

- ওয়েবসাইটে অ্যাড টু কার্ট থেকে কিনতে পারবেন, তাছাড়াও রেজিস্ট্রেশন ফর্মের মাধ্যমে ম্যানুয়ালিও কিনতে পারবেন।

ওয়েবসাইট লিংক: **এখানে ক্লিক করুন**

গুগল রেজিস্ট্রেশন ফর্ম: **এখানে ক্লিক করুন**

✦ কীভাবে পেমেন্ট করবো?

- আমাদের বিকাশ মার্চেন্ট নাম্বার: BKash : 01977659043, Bkash: 01798442589, Merchant Name: Interactive Ventures Ltd. ওয়েবসাইট থেকে কিনলে ভিসা কার্ড, মাস্টার কার্ড, বিকাশ, যে কোন মাধ্যমেই পেমেন্ট করা যাবে। এছাড়াও আপনি দেশের বাইরে থেকে পেমেন্ট করতে চাইলে Stripe -এ পেমেন্ট করতে পারবেন।

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

✦ কোর্সে ভর্তি বাতিল করা কী সম্ভব?

- না, কোর্সে এনরোলমেন্টের পরে তা আর বাতিল করার অপশন নেই।

✦ কোন টেকনিকাল সমস্যা কীভাবে রিপোর্ট করবো?

- ওয়েবসাইটের হেল্প অপশনে টেক্সট দিতে পারেন কিংবা আমাদের ফেইসবুক পেইজে আমাদের রিচ করতে পারেন। এছাড়াও যেকোনো প্রয়োজনে 01958-622164 এই নাম্বারে কল করুন।

✦ কোর্স কেনার পর রিফান্ড পাবো কীভাবে?

- না, কোর্স কিনে ফেলার পরে আমরা কোনো রিফান্ড অপশন অফার করছি না।

GOLANG
BACKEND
DEVELEPMENT
MASTERY
BATCH 1

Interactive
CARES

প্রোডাকশন-গ্রেড Backend তৈরি করুন Golang দিয়ে - এনরোল করুন আজই!

আপনি যেকোনো প্রোগ্রামিং ব্যাকগ্রাউন্ড থেকে আসুন, এই কোর্সে শিখবেন
production-ready backend তৈরি করার স্কিল।
আপনার ব্যাকএন্ড স্কিলকে নেক্সট লেভেলে নিয়ে যান এখনই।

যেকোনো সমস্যায় পড়লে

কল করুন

আমাদের কাস্টমার কেয়ার নাম্বারে

+880 1958622164

শেখা শুরু করি

কমিউনিটিতে জয়েন করি

Course Outline

Golang Backend Development - 40-Day Complete Course Plan

From fundamentals → microservices → gRPC → pub/sub → Temporal → geospatial → operators → industry-grade capstones (ride-sharing). Includes day-by-day objectives, labs, deliverables, and acceptance criteria.

Phase 0: Environment Setup (Pre-Day)

- Install: Go, Docker, docker-compose, buf/protoc, k6, Temporal CLI.
- Create monorepo skeleton; set up libs/config, libs/logger, libs/errors, libs/tracing.
- Bring up local infra with docker-compose: Postgres, Redis, Jaeger.
- Acceptance: docker compose up starts infra; sample HTTP service runs and is traceable in Jaeger.

Phase 1: Golang Fundamentals (Days 1–8)

Day 1 - Language Basics

- Topics: syntax, variables, types, control flow, modules.
- Lab: CLI parses flags, reads env, prints structured JSON.
- Deliverable: tools/cli-basics with README.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Day 2 — Interfaces, Methods, Errors

- Topics: interfaces, methods, error handling patterns, package layout.
- Lab: libs/errors with sentinel errors and wrapping; libs/config (env + defaults).
- Deliverable: library tests demonstrating error wrapping and config precedence.

Day 3 — Data Structures & Generics

- Topics: arrays, slices, maps, structs, generics (Go 1.18+).
- Lab: generic Set[T], RingBuffer[T].
- Acceptance: 90% test coverage; benchmarks for RingBuffer.

Day 4 — Build Custom DS

- Topics: heap, queue, linked list, graph.
- Lab: min-heap, FIFO queue, doubly linked list, adjacency-list graph.
- Acceptance: tests + benchmarks; no race detector warnings.

Day 5 — Concurrency I

- Topics: goroutines, channels, select, context cancellation.
- Lab: worker pool + fan-out/fan-in; graceful shutdown.
- Acceptance: controlled concurrency with context; unit tests simulate cancellation.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Day 6 — Concurrency II Project

- Project: concurrent web scraper with rate limits and retries.
- Acceptance: configurable concurrency; backoff; CSV output; idempotent runs.

Day 7 — Testing, Benchmarking, Profiling

- Topics: table tests, testify, pprof, race detector.
- Lab: profile DS/worker pool, remove hotspots.
- Acceptance: flamegraph analysis documented; 20% improvement over baseline.

Day 8 — HTTP Server Basics

- Topics: router, middleware, auth, graceful shutdown.
- Project: simple HTTP server with logging, recovery, auth middleware.
- Acceptance: health endpoint, structured logs, OpenAPI stub.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Phase 1.5: Backend Service Mini-Sprint (Days 9–10)

Day 9 — User Service + Driver Service (REST) We will build two microservices in this module user-service which holds and act like a user-service in concept of ride-sharing app

- User Service
  - Endpoints: POST/GET/PATCH/DELETE /users, GET /users/{id}
  - Fields: id, name, email, phone, auth_hash, status, created_at
  - JWT auth with password hashing; validation; rate limiting
  - Storage: Postgres + migrations; Redis cache for hot

- Driver Service
  - Endpoints: POST/GET/PATCH/DELETE /drivers, GET /drivers/{id}
  - Fields: id, name, phone, vehicle, license, rating, availability_status, current_location (lat/lng, geohash), created_at
  - Redis GEO for proximity; background simulated heartbeat updater
  - Elasticsearch based realtime driver look up service for ride sharing app

- Acceptance: both services run; JWT-protected endpoints; traces visible across both.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Day 10 — OMS (Order Management System) + Basic API Gateway

- OMS Service
  - Purpose: manage ride orders state machine: requested → searching → assigned → en_route → completed | canceled
  - Endpoints: POST /rides, GET /rides/{id}, PATCH /rides/{id}/state, GET /rides
  - Idempotency keys on create/update; validate state transitions
  - References: user_id, driver_id (service-to-service REST for enrichment)
  - Emits domain events (in-memory or webhook stub) for later pub/sub
- API Gateway (basic)
  - Routes: /users → User; /drivers → Driver; /rides → OMS
  - Adds request ID, rate limits, auth verification
- Load Gen: k6 scripts for creating users/drivers/rides and transitioning states.
- Acceptance: E2E flow works: create user → driver → ride → transition states; distributed trace across services.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Phase 2: Microservices Foundations (Days 11–12) — before gRPC

Day 11 — Service Design & Data Ownership

- Concepts: bounded contexts, API design, pagination/filtering, idempotency, outbox pattern, eventual consistency.
- Sync vs async; retries/timeouts; circuit breaker/bulkheads.
- Hands-on: add request timeouts/retries and a circuit breaker on OMS → Driver lookups; introduce outbox table in OMS (no bus yet).
- Acceptance: chaos test (fail Driver intermittently) shows graceful degradation and retry behavior.

Day 12 — Reliability, Contracts, Observability, Security

- Contracts: OpenAPI, semver, deprecation policy, compatibility testing.
- Observability: OTel traces/metrics/logs, correlation IDs; SLOs/SLIs (assign latency, success rate).
- Security/config: JWT scopes, secrets management, config layering.
- Hands-on: add golden path dashboards and alerts; add contract tests for APIs.
- Acceptance: dashboards for OMS errors/latency; alert fires on 5xx spike; contract tests pass.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Phase 3: gRPC & Interservice Communication (Days 13–16)

Day 13 — Protobuf & gRPC Fundamentals

- Protobuf syntax, unary/streaming, status codes; buf lint/build.
- Deliverable: convert User Service to gRPC with deadlines and TLS (dev).
- Acceptance: gRPC client with deadline and retry policy succeeds under injected latency.

Day 14 — Interceptors & REST Gateway

- Interceptors: logging, metrics, auth; per-RPC credentials; tracing propagation.
- gRPC-Gateway exposing REST façade.
- Acceptance: dual-stack (gRPC + REST) for User; OpenAPI generated from proto.

Day 15 — Service Discovery, Load Balancing, Cancellations

- Client-side LB, retry budgets, timeouts; propagate context.Context cancellation.
- Deliverable: Orders gRPC → Catalog gRPC (sample) with cancellation demo.
- Acceptance: cancellation propagates and stops work; metrics reflect cancellations.

Day 16 — E2E gRPC Integration

- Convert Driver or OMS to gRPC for critical paths; keep REST for external.
- Acceptance: mixed-mode interservice comms with tracing across protocols.

Phase 4: Temporal Fundamentals & Workers (Days 17–22)

Day 17 — Temporal Core

- Architecture, server, workers; workflows vs activities; determinism.
- Deliverable: local Temporal stack; Hello Workflow; Web UI exploration.
- Acceptance: first workflow deploy and execution visible.

Day 18 — Workers, Retries, Heartbeats

- Worker lifecycle; activity retries/timeouts; heartbeats; cancellation.
- Project: file processing worker (checksum, store, mark complete).
- Acceptance: simulated long-running activity shows heartbeat and cancel.

Day 19 — Workflow Composition

- Signals, queries, timers, child workflows, selectors; parallel fan-out.
- Lab: parallel ExecuteActivity with selectors and timeout guards.
- Acceptance: deterministic behavior under replays; unit tests for signals/queries.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Day 20 — Sagas & Idempotency

- Compensation, saga pattern; idempotency keys; dedup strategies.
- Project: order processing workflow (reserve inventory → payment pre-auth → confirm).
- Acceptance: injected failures trigger compensations correctly.

Day 21 — Workflow Versioning & Updates

- Versioning APIs; backward compatibility; migration strategies.
- Acceptance: deploy a versioned update without breaking in-flight runs.

Day 22 — Temporal Observability

- Metrics, trace links, DLQ handling, replay tooling.
- Deliverable: Temporal dashboards + DLQ replay CLI.
- Acceptance: DLQ scenario replayed and resolved.

Phase 5: Pub/Sub + Temporal Pipeline (Days 23–25)

Day 23 — Messaging Design

- Kafka/Redpanda vs Redis Streams vs GCP Pub/Sub; topics/partitions/keys; ordering; consumer groups; DLQ.
- Design topics: orders.v1, payments.v1, rides.v1; DLQ strategy; idempotency store.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Day 24 — API → Pub/Sub → Consumer → Temporal

- API publishes order/ride events to topic.
- Stateless consumer pushes work to Temporal:
  - Activities: payment pre-auth, inventory/driver hold; child workflow: scheduling/assignment.
- Observability: trace propagation from API → consumer → workflow → activities.
- Acceptance: idempotent consumer; at-least-once with dedup; DLQ wired.

Day 25 — Backpressure & Throughput

- Backpressure, batching, exactly-once-ish semantics; partitioning by user/region/geohash.
- Load test with k6; document autoscaling strategies (HPA/KEDA).
- Acceptance: meet baseline TPS with bounded p99 latency; DLQ remains near zero under normal load.

Phase 6: System Design & Geo/AB/Observability (Days 26–30)

Day 26 — Ride Sharing Architecture

- Services: Rider, Driver, OMS, Matching, Pricing, ETA, Payment, Notification.
- Contracts: APIs/schemas/topics; SLIs/SLOs; privacy and PII concerns.
- Acceptance: complete architecture spec with sequence diagrams and topic schemas.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Day 27 — Geospatial Service

- Geohash, R-tree/quadtree; proximity search; Redis GEO vs ES geo-indexing.
- Deliverable: geo service with Redis GEO primary and ES fallback; ingestion from driver location updates.
- Acceptance: radius search ≤ 50ms (local); fallbacks documented.

Day 28 — Matching Service

- Candidate selection; scoring; bidding window; timeouts/fallbacks.
- Temporal workflow for matching orchestration: fan-out to candidates, accept-first, compensation on timeout.
- Acceptance: simulations show stable assign latency under varying driver density.

Day 29 — A/B Testing Proxy

- Weighted routing, sticky bucketing, holdouts; experiment configs.
- Temporal workflow for experiment lifecycle (start/pause/analyze/finalize).
- Acceptance: canary deploy with AB proxy; metrics compare variants.

Day 30 — Observability Platform

- Metrics/logs/traces; SLO dashboards; alert policy for assign latency and match rate.
- Acceptance: complete dashboard set + alert rules; runbook with troubleshooting steps.

Phase 7: Advanced Temporal & Scalability (Days 31–34)

Day 31 — Priority Workers & Multi-Tenancy

- Priority queues: jobs.{priority}.{v}; weighted fair scheduling; tenant isolation.
- Acceptance: stress test shows high-priority jobs preempt low-priority within SLA.

Day 32 — Cross-Service Sagas

- Trip booking across ride, payment, notification; compensation sequences.
- Chaos testing with fault injection.
- Acceptance: recovery without orphaned states; consistency verified.

Day 33 — Long-Running Activities

- Route precompute; heartbeats; resumable execution; cancellation tokens.
- Acceptance: pause/resume workflow mid-activity; state consistent.

Day 34 — Autoscaling & Production Hardening

- KEDA/HPA on Kafka lag; resource requests/limits; Temporal cluster HA; retention/quotas.
- Acceptance: scaling policy manifests; capacity plan; failure drill.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Phase 8: Infra, CI/CD, K8s (Days 35–37)

Day 35 — Redis Deep Dive

- Geo sets for proximity cache; rate limits (token bucket); cache-aside/write-through/TTL.
- Acceptance: proximity cache integrated; per-rider/driver rate limits enforced.

Day 36 — Elasticsearch/Quickwit

- Indexing, analyzers, aggregations, ILM; reindex workflows via Temporal.
- Acceptance: trip search with multi-tenant indices; reindex workflow demo.

Day 37 — Docker/CI/Kubernetes

- Multi-stage Dockerfiles; SBOM; CI pipeline (lint/test/build/scan).
- Kubernetes: deploy Temporal + services; blue/green + canary via A/B proxy; HPA/KEDA.
- Acceptance: one-click deploy to Kind; passing CI; smoke tests green.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Phase 9: Capstones (Days 38–40)

Days 38–39 — Capstone A: Ride Sharing Backend

- Services: Rider, Driver, OMS, Matching, Pricing, ETA, Payment, Notification, Geo, WS Gateway.
- Topics: driver.location.updates, trip.requested/assigned/started/completed, payment.auth/capture/refund.
- Temporal:
  - Matching orchestration (candidates, bidding window, fallback).
  - Payment workflows (auth/capture/refund).
  - Driver shift/availability lifecycle.
- Scaling:
  - WebSocket gateway for live locations.
  - Partitioning by geohash/region.
  - Autoscaling workers by topic lag.
- Observability: end-to-end traces; SLO dashboards; alerts.
- Acceptance: load tests meet SLOs; rollback drill successful.

Day 40 — Finalization & Productionization

- System demo, architecture doc, runbooks, incident simulation & postmortem.
- Cleanup: backlogs, DLQs, dashboard annotations, cost notes.
- Acceptance: full demo recorded; documentation complete and reviewed.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Optional Alternate Capstones

- Food Delivery/Fleet Management (waves/rebalancing workflows).
- Pub/Sub Search Ingestion: API → Kafka → Temporal → ES/Quickwit (reindex + schema migration workflows).
- Scalable WebSocket Pub/Sub: presence, rooms, Redis Pub/Sub; Temporal for moderation/retention.

Milestones (Weekly)

- Week 1 (Days 1–8): Fundamentals complete; DS & concurrency labs benchmarked.
- Week 2 (Days 9–16): Core services (User/Driver/OMS) + Microservices Foundations + gRPC.
- Week 3 (Days 17–22): Temporal basics; file + order workflows; versioning; observability.
- Week 4 (Days 23–30): Pub/Sub pipeline; ride sharing design; geo/matching/AB/observability.
- Week 5 (Days 31–34): Advanced Temporal patterns; autoscaling; hardening.
- Week 6 (Days 35–37): Infra/CI/K8s; deploy to Kind.
- Week 7 (Days 38–40): Capstone implementation and demo.

GOLANG BACKEND DEVELEPMENT MASTERY BATCH 1

Interactive CARES

Module-wise Outcome

- Golang fundamentals ও concurrency নিয়ে হ্যান্ডস-অন এক্সপেরিয়েন্স
- REST & gRPC microservices design, development এবং integration
- Temporal workflows, Pub/Sub pipelines এবং reliable event-driven architecture তৈরি করা
- System design, observability, autoscaling এবং production-grade backend প্রজেক্টে অ্যাপ্লাই
- CI/CD, Docker, Kubernetes deployment ও real-world distributed systems চালানো

GOLANG
BACKEND
DEVELEPMENT
MASTERY
BATCH 1

Interactive
CARES

প্রোডাকশন-গ্রেড Backend তৈরি করুন Golang দিয়ে - এনরোল করুন আজই!

আপনি যেকোনো প্রোগ্রামিং ব্যাকগ্রাউন্ড থেকে আসুন, এই কোর্সে শিখবেন
production-ready backend তৈরি করার স্কিল।
আপনার ব্যাকএন্ড স্কিলকে নেক্সট লেভেলে নিয়ে যান এখনই।

যেকোনো সমস্যায় পড়লে

কল করুন

আমাদের কাস্টমার কেয়ার নাম্বারে

+880 1958622164

শেখা শুরু করি

কমিউনিটিতে জয়েন করি