IRCTC-Style Ticket Booking System (Go Backend)

This document explains which technology is used for which functionality and provides a conceptual architecture layout for building a high - concurrency IRCTC - like booking system.

---

1. API Backend – Go (Golang)

Responsibilities:
- Exposes REST APIs (search trains, lock seats, book ticket, pay).
- Implements business rules.
- Communicates with PostgreSQL and Redis.
- Completely stateless → allows horizontal scaling.

Why Go?
- Excellent concurrency (goroutines + channels).
- Very fast and lightweight.
- Perfect for high - load booking systems.

---

2. PostgreSQL – Transactional Storage

Used for:
- Permanent booking records.
- Train, seat, coach metadata.
- User accounts.
- Payment status.
- Audit logs.

Why PostgreSQL?
- Strong ACID guarantees → prevents double booking.
- Supports transactions and row - level locking.
- Enforces constraints such as UNIQUE(seat_id, date).

---

3. Redis – Concurrency Control Layer

Used for:
A) Distributed Seat Locking
- When a user selects a seat, Redis performs atomic lock:
SET seat:{train}:{seat} value EX 120 NX

B) Seat Hold TTL (Temporary Reservation)
- Locks automatically expire after a fixed timeout.

C) Tatkal Queue Handling
- Redis streams or queues buffer millions of 10AM requests.
- Protects database from overload.

D) Rate Limiting
- Token bucket / sliding window logic.

Why Redis?
- Extremely fast in - memory performance.
- Atomic operations with low latency.
- Built for concurrency.

---
4. BullMQ / Redis Stream Workers

Used for:
- Tatkal booking processing.
- Payment confirmation.
- Releasing expired locks.
- Sending booking notifications.

Why?
- Booking workflow must be asynchronous.
- API should return fast; workers handle heavy processing.

---
5. Nginx / API Gateway

Responsibilities:
- HTTPS termination.
- Load balancing to Go backend instances.
- Request throttling.
- Routing.

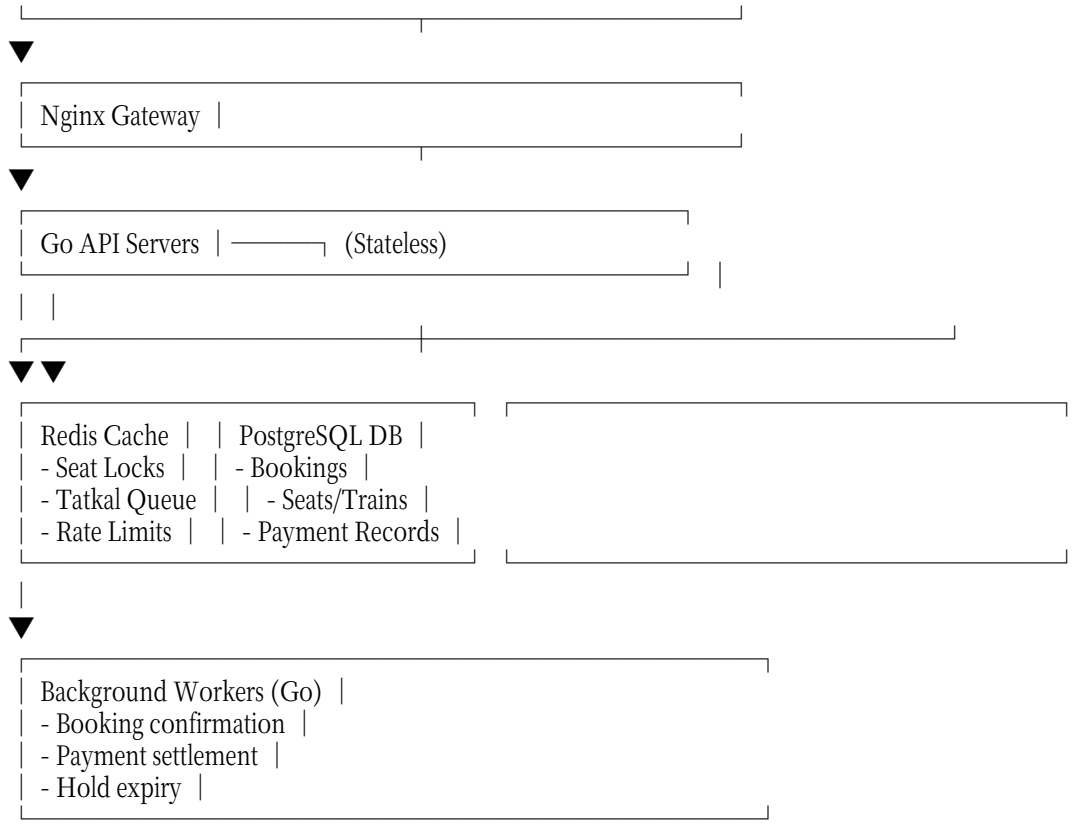
---
6. Prometheus + Grafana – Monitoring

Used for tracking:
- API request rate (RPS).
- Queue size.
- Lock failures.
- DB performance.
- Worker throughput.

---
7. k6 / JMeter – Load Testing Tools

Used for:
- Simulating Tatkal traffic (mass concurrency test).
- Testing seat - locking correctness under heavy load.
- Payment spike simulation.

Architecture Diagram (Textual Representation)



This high - level architecture ensures:
- No double booking.
- Smooth handling of Tatkal load.
- Horizontal scalability.
- Clear separation of responsibilities.