Tier 0 — Non-Negotiable (platform cannot function without these)

#	Algorithm	What it does	Why it matters	Typical method
1	**Rider Candidate Filtering**	Builds the set of riders eligible for an order (online, within radius, capacity, verified)	Garbage in → garbage out for dispatch	Geofencing (Haversine), rule filters
2	**Rider Dispatch & Selection (Scoring)**	Scores riders and selects best one; sequential job offers	Core marketplace engine	Weighted scoring function
3	**ETA Estimation (Pickup & Dropoff)**	Predicts arrival times using distance, speed, prep time	Customer trust, reduces cancellations	Regression / heuristic speed profiles
4	**Route Planning / Shortest Path**	Computes optimal route between points	Directly impacts delivery time	Dijkstra / A* on road graph
5	**Delivery Fee Pricing**	Calculates delivery charge from distance, time, zone	Revenue correctness	Rule engine + distance bands
6	**Order State Machine**	Controls valid order transitions	Prevents logic bugs & fraud	Finite State Machine (FSM)

Tier 1 — Reliability & Scalability (makes system stable)

#	Algorithm	What it does	Why it matters	Typical method
7	**Shop Prep Time Estimation**	Predicts how long a shop takes to prepare	Avoids rider waiting	Historical averaging / regression
8	**Reassignment / Timeout Logic**	Reoffers job if rider fails to accept/act	Prevents stuck orders	Timers + fallback queue
9	**Batching / Order Stacking**	Assigns 2+ orders to one rider if routes align	Efficiency gain	Greedy route merge / VRP-lite
10	**Service Area Zoning**	Divides city into operational zones	Enables dispatch control	Grid / H3 hex clustering
11	**Hotspot Detection**	Detects high demand zones over time	Rider positioning, surge	Kernel Density / grid counts
12	**Supply Forecasting**	Predicts rider availability per zone/time	Balance demand vs supply	Time-series averages
13	**Fraud / Abuse Detection**	Detects suspicious rider/customer/shop behavior	Protects margins	Rule engine + anomaly scoring

Tier 2 — Competitive Advantage

#	Algorithm	What it does	Why it matters	Typical method
14	**Dynamic Surge Pricing**	Raises fees when demand > supply	Keeps service reliable	Zone load ratio rules
15	**Rider Repositioning Recommendation**	Suggests where riders should wait	Improves earnings & coverage	Hotspot + rider density logic
16	**Personalized Shop/Item Ranking**	Ranks shops for each customer	Increases conversions	Collaborative / content filtering
17	**Cancellation Risk Prediction**	Predicts likelihood of order cancel	Prevent wasted effort	Classification model
18	**QoS Optimization (Multi-Objective)**	Balances fairness, ETA, profit, acceptance	Advanced fleet control	Weighted optimization

Tier 3 — Advanced Optimization / Research Level (later stage)

#	Algorithm	What it does	Why it matters	Typical method
19	**Vehicle Routing Problem (VRP)**	Optimal multi-delivery routing with constraints	Large efficiency gains	VRP / Pickup-Delivery solver
20	**Contextual Bandits for Dispatch**	Learns best dispatch choices over time	Data-driven improvement	Multi-armed bandits
21	**Reinforcement Learning Fleet Control**	Learns surge, incentives, positioning	Marketplace automation	RL policy learning

Mapping & Geospatial Support Algorithms (used by many above)

Algorithm	What it does	Used by
Haversine Distance	Fast distance between lat/long	Filtering, pricing, ETA
Map Matching	GPS → likely road path	ETA, fraud detection
Geocoding / Reverse Geocoding	Address ↔ coordinates	Orders, shops, riders
Road Speed Profiling	Learns speeds per road/time	ETA accuracy
Hex/Grid Spatial Indexing (H3)	Spatial clustering of city	Zoning, hotspots, surge

First 5 You Should Implement (true v1 core)

1. Rider Candidate Filtering
2. Rider Dispatch & Selection (Scoring)
3. ETA Estimation
4. Route Planning (Shortest Path)
5. Order State Machine