GEARFALL — Codex Build Bible (Web-First → Mobile Migration)

Version 1.0 • February 28, 2026 • Combat, Loot, Economy, API, Telemetry, UI, Runbook

0. How to Use This With Codex (Images-Only Attachments)

You will upload the rendered PNG pages of this document into your Git repo so Codex can reference them. This build bible is implementation-ready: contracts, algorithms, schemas, and acceptance criteria.

Recommended workflow:

- Commit the DOCX to /docs/spec/Gearfall_Build_Bible.docx (for humans).
- Commit the PDF to /docs/spec/Gearfall_Build_Bible.pdf.
- Commit the PNG pages to /docs/spec/images/ (Codex-friendly).
- In Codex prompt, instruct it to read /docs/spec/images and follow the spec verbatim.

1. Game Identity

Gearfall is a mech tactical RPG where battles produce recoverable technology ('gearfall'). Players salvage randomized mech components, optimize builds via affixes, and climb encounter tiers.

1.1 Pillars

- Hybrid ATB turn-based encounters (tactical decisions, short sessions).
- Diablo-style randomized loot (rarity + affixes + procs).
- Modular mech loadouts (slots, set bonuses later).
- Server-authoritative progression + config-driven economy.
- Telemetry-first iteration (retention + monetization tuning).

2. MVP Scope (Phase 1)

- Mobile-first PWA shell: bottom tabs, full-screen pages, sheet modals, 44px tap targets.
- Encounter loop: startEncounter → resolve turns → completeEncounter → rewards reveal.
- Loot generation: rarity + affixes + item power; deterministic RNG per encounter seed.
- Gear inventory + equip/unequip + upgrade (basic).
- Energy system (cap + regen) + daily missions (3).
- Offer framework (view, eligibility, stub purchase).
- Telemetry ingestion + DB storage + docs.

3. Repo Structure

```
repo-root/
  apps/
```

```
  web/
  api/
packages/
  core/
  ui/
infra/
  docker/
docs/
  spec/
    Gearfall_Build_Bible.docx
    Gearfall_Build_Bible.pdf
    images/
  architecture.md
  api.md
  telemetry.md
  runbook.md
```

4. Combat System (Hybrid ATB)

4.1 Stats (MVP)

Stat	Description
HP	Hit points
ATK	Attack power
DEF	Defense
SPD	Turn meter fill rate
CRIT%	Critical chance
CRITx	Critical multiplier
SHD	Shield capacity

4.2 Turn Meter

```
meter += SPD * tickScalar
if meter >= 100:
 takeTurn()
 meter -= 100
```

4.3 Damage Formula (Server-Side)

```
base = max(1, ATK * skillMultiplier - DEF * defenseScalar)
if rand() < CRIT_CHANCE:
 base *= CRIT_MULT
# shields first
dS = min(SHD, base)
SHD -= dS
```

```
HP -= (base - dS)
```

5. Loot & Gear (Diablo-Style)

5.1 Slots

Slot	Primary Bias
Weapon System	ATK/CRIT/Procs
Armor Plating	DEF/HP/DR
Power Core	Energy/SHD/Regen
Mobility System	SPD/Evasion
Tactical Module	CDR/Utility

5.2 Rarity Rules

Rarity	Weight	Affixes
Common	70%	0–1
Rare	25%	1–2
Epic	4.5%	2–3
Legendary	0.5%	3–4 + Proc

5.3 Affix Pool (MVP)

Affix	Slots	Range
+HP	Armor/Core	100–300
+ATK	Weapon	10–35
+DEF	Armor	5–20
+SPD	Mobility/Module	1–6
+CRIT%	Weapon/Module	1–8%
Cooldown Reduction	Module	1–10%
Energy Regen/Turn	Core	1–3

5.4 Legendary Proc Pool (MVP)

Proc	Trigger	Effect
EMP Burst	On Crit	+true dmg; -1 enemy SPD (1 turn)
Overcharge	On Ability	Next basic +% dmg
Shield Siphon	On Kill	Gain +shields

5.5 Loot Generation Algorithm

```
Deterministic server loot (seeded):
1) Roll rarity by weights (tier-modified).
2) Roll slot by table weights.
3) Pick base template.
4) Roll itemPower scalar from rarity range and tier scaling.
5) Roll affix count; sample affixes allowed for slot; roll values.
```

```
6) If Legendary, roll 1 proc.
7) Persist GearInstance with source {encounterId, seed, tableVersion}.
```

6. Economy & Live Ops

6.1 Energy

```
energy.max = 100
energy.regenPerMinute = 1
encounter.cost = 10
```

6.2 Daily Missions (3)

Mission	Trigger	Reward
Complete 3 encounters	encounter_complete x3	Gold+Scrap
Upgrade gear 1 time	upgrade_success x1	Gold
View shop	offer_view x1	Energy

6.3 Offers (MVP)

- Starter Pack (one-time; after first win).
- Energy Pack (repeatable).
- Limited Pack (timer; after loss or tier unlock).
- Phase 1 purchase is stubbed but grants entitlements and records telemetry.

7. API Contract (Encounter-Based)

```
Auth: POST /auth/dev-login
Player: GET /player/me, POST /player/reset

Encounter:
- POST /encounter/start -> { encounterId, seed, config }
- POST /encounter/action -> { encounterId, action } -> { resolutionStep }
- POST /encounter/complete -> { rewards, newState }

Gear:
- GET /inventory
- POST /gear/equip
- POST /gear/upgrade

Offers:
- GET /offers
- POST /offers/:id/view
- POST /offers/:id/purchase (stub)
```

```
Missions:
- GET /missions/today
- POST /missions/:id/claim

Telemetry:
- POST /events (batch)
```

8. Telemetry (Must Implement)

Event	Key Fields
app_open	sessionId, deviceMeta
screen_view	screenName
encounter_start	tier, energyBefore
encounter_action	actionType
encounter_complete	result, durationMs
reward_reveal	rarityCounts
gear_equip	gearId, slot
upgrade_success	gearId, level
offer_view	offerId
purchase_stub	offerId, price
mission_claim	missionId

9. Mobile-First UI Rules

- Bottom tabs: Home, Encounter, Mech, Shop, Missions.
- Sheet modals for details; full-screen pages.
- 44px min tap targets; safe-area insets; no hover interactions.

10. Security & Integrity

- Server authoritative; no client stat deltas.
- Idempotency for completion and purchase endpoints.
- Store seed proof for loot; write ledger for all deltas.
- Rate limiting on auth/encounter endpoints.

11. Runbook (Commands)

```
pnpm install
docker compose up -d
pnpm --filter api db:migrate
pnpm --filter api db:seed
pnpm --filter api dev
pnpm --filter web dev
```

12. Codex Master Prompt (Images-First)

Read the spec PNGs in /docs/spec/images.

Build Gearfall as a mobile-first PWA with encounter-based hybrid ATB combat and randomized mech gear.
Server-authoritative, deterministic loot seeded per encounter, config-driven economy, telemetry, docs, CI.
Follow the repo structure and UI rules exactly.