

## GROWNET — Codebase & Engineering Quality Pack

---

### 1. Purpose

هدف این سند:

نشان دادن اینکه کد base طوری طراحی شده که قابل نگهداری، قابل تحویل مداوم و مقاوم در برابر ریسک تیمی باشد.  
ما دنبال کد آکادمیک نیستیم؛ دنبال قابل تکرار delivery هستیم.

---

### 2. Repository Overview

#### Repo Structure (Example)

/apps

/web

/api

/packages

/auth

/ranking

/billing

/shared

/infrastructure

/terraform

/ci

/docs

#### Why this structure?

تفکیک concern •

آماده برای scale •

جلوگیری از spaghetti code •

---

### 3. Coding Standards & Conventions

#### Language & Stack

Backend: Node.js / Python •

Frontend: React •

Infra: Docker + Terraform •

Standards	
ESLint / Prettier	•
Commit message convention	•
Code review mandatory (PR-based)	•
هیچ commit main نداریم. مسنتقیم به	

---

#### 4. Testing Strategy

Type	Coverage	Tool
Unit	Core logic	Jest / PyTest
Integration	API flows	Supertest
E2E	Critical paths	Playwright
<b>Rule:</b> No merge without green tests.		

---

#### 5. CI/CD Pipeline

CI Steps:	
Lint	.1
Test	.2
Build	.3
Security scan	.4
CD:	
Auto deploy to staging	•
Manual approval to prod	•
Rollback enabled	•
هدف deploy: بدون ترس	

---

#### 6. Versioning & Release Management

Semantic versioning	•
Release notes mandatory	•
Feature flags for risky changes	•

---

## 7. Technical Debt Register

Debt	Impact	Plan
Hardcoded configs	Medium	Env-based config
Ranking logic coupling	High	Isolate service
Missing load tests	Medium	Add before scale
دیده شده، نه انکار شده		debt

## 8. Documentation & Knowledge Transfer

### Docs Included:

- Architecture overview •
- API contracts •
- Setup guide •

### Knowledge Risk Mitigation:

- Onboarding checklist •
- Pair programming •
- Recorded walkthroughs •

اگر یک نفر برود، سیستم نمی‌ریزد

## 9. Bus Factor Mitigation

Risk	Mitigation
Key dev leaves	Shared ownership
Solo modules	Mandatory rotation
Tribal knowledge	Docs + diagrams

## 10. Security & Code Quality

- Secret management •
- Dependency scanning •
- Static analysis •

## 11. Engineering KPIs

Metric	Target
Lead time	<7 days
Deployment freq	Weekly
Change failure	<10%
MTTR	<1h

---

## 12. Engineering Culture

- Quality over speed (but no paralysis)
  - Fix root cause, not symptom
  - Continuous refactor windows
- 

## Investor / CTO Takeaway

«این تیم کدی می‌نویسد که قابل تحویل، قابل نگهداری و قابل انتقال است؛ حتی اگر تیم تغییر کند».