# ✅ Final English Prompt

**You are a senior front-end engineer. Build a fully functional, single-file, demo-ready prototype of a Persian (Farsi) Open Innovation Platform modeled after InnoCentive and NineSigma, PLUS a dedicated TESTA exhibition module.** **Constraints:** All UI and text must be RTL (Persian-first). Use React (hooks) + Tailwind via CDN (no build step). Mock all backend behavior with in-memory state and localStorage. Deliver everything as a single index.html (one <script type="text/babel"> with one exported App component and subcomponents inside the same file). No external tooling required to run.

## **0) Goals & Audience**

* Demonstrate a year-round **Open Innovation Platform** connecting **Seekers** (organizations with challenges) to **Solvers** (innovators), **plus** an **Exhibition (TESTA)** section with time-boxed pages (schedule, events, map, announcements, QR access, kiosk mode).
* The prototype must be **clickable end-to-end**: register/login (demo accounts), post challenges, submit confidential solutions (NDA gate), evaluate & select winner, message privately, and show TESTA exhibition info.

## **1) Roles, Auth, and Demo Data (mocked)**

* Roles: ADMIN, SEEKER, SOLVER.
* Demo accounts:  
  + Admin: admin@platform.ir / 123456
  + Seeker: info@parstech.ir / 123456
  + Solver: a.mohammadi@university.ac.ir / 123456
* Registration flows:  
  + **Seekers**: organizational registration request (stored as pending until Admin “approves” in UI).
  + **Solvers**: individual/team registration (basic verification toggle by Admin).
* On login, set role in mock session (localStorage) and show role-aware nav.
* Profiles:  
  + Solver profile: name, bio, expertise tags, links (LinkedIn/GitHub), portfolio list, **reputation** number.
  + Seeker org profile: name, logo (upload simulated), domains/industries, about, members (add/remove via UI).
* Reputation events (mock): profile completed (+5), on-time submission (+2), accepted solution (+20).

## **2) Global UX & Tech**

* **Persian-first:** <html dir="rtl" lang="fa">, Persian fonts (system stack) and proper numerals where relevant.
* **Tailwind via CDN only**; no build steps. Keep styling clean and professional.
* **Routing:** simple hash-router or internal state tabs (no external libs).
* **Persistence:** in-memory arrays; mirror to localStorage for reload resilience; **Reset Demo** button clears state.
* **NDA gate:** modal that must be accepted before accessing submission form or sensitive attachments; store timestamp/version.
* **Notifications:** toasts for key events (submission received, status changed).
* **Validation:** real-time form validation with clear Persian messages.
* **Accessibility:** semantic markup, keyboard focus, high contrast.

## **3) Core Platform (year-round)**

### **3.1 Public (unauthenticated)**

* **Landing (خانه):** value proposition, CTAs (“ثبت چالش” for Seekers, “یافتن چالش‌ها” for Solvers), quick links to TESTA section.
* **Challenges catalog:** searchable/filterable list; cards show title, brief, category, deadline, reward, Seeker org; sensitive files hidden until login & NDA.
* **Challenge detail:** full brief, success criteria, deliverables, reward, deadline, org info; “حل این چالش” button → login/nda/submit.

### **3.2 Solver portal**

* **Dashboard:** suggested challenges (by tags), my stats, latest notifications.
* **My Submissions:** list with status chips: Received, Under Review, Needs Revision, Accepted, Rejected; can edit/withdraw pre-deadline.
* **Submit Solution form:** executive summary, detailed proposal, files (simulate upload), demo video link, innovation highlights. Show deadline and countdown. Enforce NDA.

### **3.3 Seeker portal**

* **Org profile:** logo (upload simulated), domains, about, members management.
* **Create/Edit Challenge:** title, problem, success criteria, deliverables, reward (amount & collaboration note), deadline, tags, attachments, explainer video.
* **Seeker dashboard:** for each challenge see submissions table with filters, quick statuses, and comparison view (side-by-side).
* **Evaluation:** rubric with fixed weights (fit 40, feasibility 25, cost/time 20, innovation 15) → total score; change submission state; **select winner** (notify solver).

### **3.4 Admin panel**

* **Admin dashboard:** counts (users, orgs, challenges by state, submissions by state), sparkline charts.
* **Approvals:** verify Seeker orgs, approve Solvers, moderate challenges (draft → pending\_review → published) and content flags.
* **Global tools:** broadcast message (toast to all), export mock CSV (download data URL).

### **3.5 Messaging**

* **Private thread per submission** (Seeker ↔ Solver ↔ Admin oversee). Newest first, simple composer.

## **4) TESTA Exhibition Module (time-boxed, but platform runs year-round)**

Create a distinct section (menu: **«نمایشگاه تقاضای ساخت و تولید ایرانی (تستا)»**) with these pages and badges:

1. **TESTA Home:** banner, **countdown** to exhibition dates, highlights, CTA buttons (Schedule, Events, Map, Register, Scan QR).
2. **Schedule (برنامه زمانی):** day-by-day agenda (keynotes, panels, workshops), filters by day/track; add to personal agenda (local state).
3. **Events (رویدادها):** cards with time, speakers, abstract; “Register” (capture name/email/role; store locally).
4. **Map & Booths:** static hall map image placeholder + list of booths/exhibitors with categories and booth codes.
5. **Announcements:** rolling bulletin for live updates during the show (admin can post in demo).
6. **Statistics (نمای نمایشگاه):** live-style charts (mock) for visitors today, scans, sessions started; tie to platform activity counts.
7. **QR Access:** generate QR codes (data-URL) for:  
   * Quick links to: Landing, Challenges, Specific Challenge, Register Event, and **Demo Login** (role preselect).
   * Include a small “How to scan” tooltip.
8. **Kiosk Mode:** toggle that:  
   * Increases UI size/contrast, hides emails partially (privacy mask), disables destructive actions, auto-logout after inactivity, **auto-reset dataset** every X minutes, and shows a floating “Reset Demo” button (admin only).
9. **TESTA-only sample content:** preloaded sample challenges from Iranian companies, seeded events, and exhibitors; show “(Demo)” watermark on sensitive UI in TESTA context.

Important: The TESTA section is **additive**. The year-round platform (challenges & submissions) must remain accessible at all times.

## **5) Sample Data (seed on first load)**

* **Challenges (pre-loaded):**
  1. Smart Energy Management System — *Pars Technology Development Company* — Reward: **50,000,000 IRR** — Category: Energy — multiple mock submissions.
  2. Interactive Online Learning Platform — *Ariana E-Learning Institute* — Reward: **75,000,000 IRR** — Category: Education — VR/AR hints.
  3. Early Disease Detection System — *Imam Khomeini Specialized Hospital* — Reward: **100,000,000 IRR** — Category: Medical — AI-based cardiac diagnosis.
* **Exhibition (TESTA) schedule/events:** 3 days, 4 tracks (Keynote/Panel/Workshop/Startup Pitches), ~12 events total with mock speakers.
* **Exhibitors & booths:** 10–20 mock rows with booth codes.
* **Demo accounts:** as above, with pre-populated org profile and solver profile.

## **6) States, Rules, and Security (mocked)**

* **Challenge states:** draft → pending\_review → published → closed → awarded/canceled.
* **Submission states:** received → under\_review → needs\_revision → accepted/rejected.
* **NDA required** before submission or file access (checkbox + modal; store consent record).
* **ACL (simulated):** Seekers see only their challenge submissions; Solvers see only their own; Admin sees all.
* **Deadlines:** disable submit/edit after deadline; award only when published & after review.
* **Reputation:** update on events (profile +5; on-time +2; accepted +20).

## **7) UI/UX Requirements**

* **Persian-first design:** full RTL, clear Persian validation and toasts.
* **Modern look:** Tailwind cards, tables, tabs, modals; responsive mobile-first; small animations OK.
* **Comparison tool:** side-by-side view of two submissions (title, summary, rubric scores, attachments list).
* **Search & Filters:** in catalog and submissions (by tag/category/status/deadline).
* **Charts:** simple SVG or lightweight inline components (no heavy libs) for TESTA stats & admin dashboard.

## **8) Interactions to Implement (happy paths)**

1. **Admin** approves Seeker & Solver; publishes a challenge.
2. **Seeker** creates a challenge; receives solver submissions; scores with rubric; sets winner.
3. **Solver** registers, accepts NDA, submits proposal with files; sees status changes; receives feedback.
4. **Messaging** between Seeker & Solver on a submission thread.
5. **TESTA visitor** opens TESTA Home, scans a QR to a specific challenge, registers for a workshop, views map & announcements.
6. **Kiosk reset** clears/reseeds all data; auto-logout after inactivity.

## **9) Implementation Details**

* Single index.html containing:  
  + Tailwind CDN, React/ReactDOM CDN, Babel standalone.
  + <div id="root"></div> and one <script type="text/babel"> exporting App.
  + All components (Nav, Forms, Modals, Tables, Charts, QR generator, Toasts) defined inside.
* **QR generation:** implement a minimal canvas-based encoder or use a tiny no-build inline script (if needed, include a small QR function inside the same file). Output data-URL <img>.
* **File uploads:** simulate with file metadata stored in state (name/size/type) and data-URL previews (no real server).
* **Export CSV:** build a data-URI from current in-memory arrays and trigger download.
* **LocalStorage keys:** users, orgs, challenges, submissions, messages, events, exhibitors, notifications, reputationEvents, settings (kiosk mode, countdown target), session.

## **10) Acceptance Criteria (must pass)**

* Runs by opening index.html in a browser (no server).
* All pages are **RTL Persian**, responsive, and navigable via top nav.
* NDA gate prevents submission/file access until accepted; consent recorded.
* Full role demo flows work (Admin/Seeker/Solver) with seeded accounts.
* TESTA pages exist (Home, Schedule, Events, Map/Booths, Announcements, Stats, QR Access) and function as described.
* Kiosk Mode works (UI adjustments, privacy mask, inactivity logout, timed auto-reset, manual Reset Demo).
* Comparison view shows two submissions side-by-side with rubric.
* Data persists across refresh (localStorage) and clears on Reset.
* No crashes; form validation messages are in Persian; toasts appear for key events.

## **11) What to Output**

* A **single index.html** file (complete) that I can save and open directly.
* Clear section comments to navigate code (e.g., // --- State Models, // --- TESTA Module, etc.).

# ℹ️ What I added/changed (and why)

1. **Dedicated TESTA module** (Home, Schedule, Events with registration, Map/Booths, Announcements, Stats, QR Access)  
    *Why:* You said the site must also serve the TESTA exhibition (time-boxed) while the platform runs year-round.
2. **Kiosk Mode + Demo Reset + Privacy Mask + Inactivity Logout** *Why:* Exhibition booths need a safe, loopable demo with minimal PII exposure and easy state reset.
3. **Countdown and QR flows** (deep links to challenge pages, role-preselected demo login)  
    *Why:* Common at exhibitions; makes it easy for visitors to jump into the right view on their phones.
4. **Side-by-side submission comparison UI** *Why:* Helps Seekers evaluate solutions quickly during/after the show.
5. **CSV export and lightweight charts** *Why:* Quick insights for Admin and TESTA “live stats” without external libs.
6. **Strict acceptance criteria & single-file delivery** *Why:* Ensures the artifact opens instantly (no build, no backend), perfect for Cursor and on-site demos.
7. **NDA gating recorded with timestamp/version** *Why:* Matches your confidentiality requirement and mimics real workflow.
8. **Seeded TESTA content (events/exhibitors) alongside platform challenges** *Why:* Demonstrates both time-boxed exhibition content and year-round challenge flow in one prototype.