JetBrains - Test Assignment

1) Brief QA Analysis

- **Business-critical flows:** Product discovery → trial/download → license/account access → purchase/renewal. Any break here hits revenue and adoption.
- **Site characteristics:** Large, content-rich, localized, frequently updated (marketing campaigns, product releases), integrates with payments, downloads/CDN, auth (JetBrains Account), analytics/consent.
- **Primary risks:** Release churn causing regressions, localization defects, broken links/assets, layout/visual drift, SEO/performance regressions, tracking/consent correctness, and checkout edge cases.

2) Key Website Areas for Priority Testing

1. **Product Pages and Downloads**

   - **Why**: These are the primary conversion points for users. Any issues with download links or product descriptions could lead to lost users.
   - **Testing Focus**: Verify download links, file integrity, and compatibility with different OS versions.

2. **User Authentication and Account Management**

   - **Why**: User accounts are essential for accessing features like documentation, subscriptions, and community forums.
   - **Testing Focus**: Login, registration, password recovery, and account settings.

3. **Checkout & Licensing**

   - **Why**: Direct revenue impact; involves many third-party/payment failure modes; tax, currency, and promo logic.
   - **Testing Focus**: Payment flow, error handling, tax and currency localization, promo code validation, and renewal processes.

4. **Localization & Content Integrity**

   - **Why**: The site supports 10+ locales, with complex date/number formats, right-to-left languages, and fallback behavior. Broken links or incorrect redirect rules can impact user experience.
   - **Testing Focus**: Validate all locales, date/number formatting, right-to-left language support, fallback behavior, and ensure all links and redirects are functional.

5. **Site Performance & SEO**

   - **Why**: Core Web Vitals, structured data, sitemaps/robots.txt, canonical/hreflang tags are critical for search traffic and conversion.
   - **Testing Focus**: Ensure fast load times, proper structured data markup, valid sitemaps, correct canonical and hreflang tags, and proper indexing by search engines.

6. **Ease of Navigation**

   - **Why**: A clear and intuitive navigation structure is essential for user experience and engagement. Poor navigation can lead to frustration and high bounce rates.
   - **Testing Focus**: Ensure menus are functional, breadcrumbs work correctly, and users can easily find content or products.

3) Recommended Manual vs. Automation Balance

- **Now (first 6–8 weeks): ~25% automation / 75% manual**
  - Rapidly automate stable, repeatable, high-value paths while manually exploring new campaigns and edge cases.
- **Steady state (after 2–3 quarters): ~80% automation / 20% manual**
  - Manual: risk-based exploratory, accessibility audits, visual review of new layouts, complex payment edge cases.
  - Automation: smoke, regression, links/assets checks, API contracts, flows for purchase → account → download.

4) Test Automation Prioritization Methodology

Use a **Risk × Frequency × Effort** model with lightweight scoring (1–5). Automate items with highest score first.

- **Risk (Impact of failure):** Revenue loss, data/privacy, brand/SEO hit, support load.
- **Frequency (How often used/changed):** Traffic %, weekly content churn, dependencies.
- **Effort (Lower is higher priority):** Test stability, data/setup complexity, mocking needs.

* example: Checkout happy-path + common variants (promo, currency, taxes) → **R=5, F=4, E=2**

5) Structured Testing Approach

A. Release/Change Flow (PR → Test env → main → production)

1. **PR (Pre-merge — usually local/dev env):**
   - Lint/build, unit/component tests (dev-owned).
   - Basic schema/hreflang validation.
2. **Test env (CI preview — dedicated/isolated deployment during merge):**
   - CI auto-deploys the change to an **disposable** or **isolated environment** with production-like config but sandboxed integrations.
   - Run **full E2E regression** with **payment sandbox**, plus core edge cases.
   - **API contract** tests; **content integrity** (sitemap, robots, broken links), locale spot-checks.
   - **Visual regression** on priority templates.
   - Capture artifacts (traces/videos, perf/a11y reports) and publish as CI attachments. If green, auto-promote to `main`.
3. **`main` (staging — broader integration & early exposure):**
   - Deploy from Test env to a **shared staging** that mirrors production infra (CDN, caching, feature flags).
   - Run the **broader integration** (E2E thin slice + live API + link/SEO checks) and **time-boxed exploratory** focused on integration risks.
   - **Data & feature flag validation**, canary config ready; optional **load/perf sample** on the Buy/Checkout funnel.
4. **Production deploy:**
   - Progressive rollout or canary + **synthetic smoke** (download, add-to-cart, checkout redirect, login) post-deploy.
   - **RUM/analytics sampling** to verify events and CWV; log/error monitoring & alerting.