**Eastern Petroleum**

**Authentication System Requirements Report**

**Purpose:**  
This report outlines the full set of non‑technical requirements and high‑level design considerations for implementing a phone‑based OTP authentication system in our Next.js + TailwindCSS website using NextAuth.js. It focuses on business goals, security standards, user experience, and operational needs rather than specific code snippets.

**1. Data Storage**

**Objective:** Securely persist user credentials and one‑time codes while ensuring scalability and reliability.

* **User Records:** Store each user’s unique identifier, contact number, verified status, and audit timestamps.
* **OTP Records:** Maintain a temporary store of generated codes with expiration metadata.
* **Technology Options:** A managed relational database (e.g., PostgreSQL, MySQL) with automated backups, or a NoSQL alternative with strong consistency guarantees.
* **Operational Requirements:**
  + Encryption at rest and in transit.
  + Automated backups and point‑in‑time recovery.
  + Retention policy for OTP entries (e.g., automatic purge after expiration).

**2. Authentication Framework**

**Objective:** Provide a robust, extensible system for sign‑in, session management, and token lifecycle.

* **Platform:** Leverage NextAuth.js to handle cookie-based sessions and token management.
* **OTP Provider:** Custom credentials flow accepting phone and OTP fields.
* **Session Strategy:** Secure, HttpOnly cookies; optional JSON Web Tokens for API calls.
* **Business Considerations:**
  + Zero downtime during rotating session secrets.
  + Clear error handling paths for expired sessions.
  + Audit logs for sign‑in attempts and session invalidations.

**3. OTP Delivery Service**

**Objective:** Ensure timely, reliable delivery of verification codes to user devices.

* **Vendor Selection:** Evaluate SMS gateways (e.g., Twilio, MSG91, Exotel) based on cost per message, delivery latency, regional coverage, and service SLAs.
* **Message Design:** Concise, branded SMS template with clear instructions.
* **Delivery Metrics:** Track delivery success rate, average latency, and failure reasons.
* **Reliability Requirements:**
  + 99.9% message delivery success.
  + Retry mechanism for transient failures.

**4. Session Management**

**Objective:** Maintain authenticated state securely across client interactions.

* **Cookie Configuration:**
  + Secure, HttpOnly, SameSite=Lax
  + Appropriate expiration consistent with OTP validity and business policy.
* **Session Renewal:** Optionally renew tokens on activity to improve user experience.
* **Logout Handling:** Immediate session invalidation and redirect to login page.

**5. API Security & Rate‑Limiting**

**Objective:** Protect authentication endpoints from abuse, fraud, and automated attacks.

* **Rate Limits:** Define thresholds (e.g., 3 OTP requests per phone number per hour; 5 failed attempts per session).
* **Monitoring & Alerts:** Implement real‑time dashboards and alerts for suspicious activity spikes.
* **Data Protection:** Store OTP codes transiently; consider hashing in persistent store.
* **Compliance:** Adhere to regional regulations for user data and telecommunication (e.g., GDPR, local telecom guidelines).

**6. Front‑End Components & UX**

**Objective:** Deliver an intuitive, accessible, and responsive user interface for authentication.

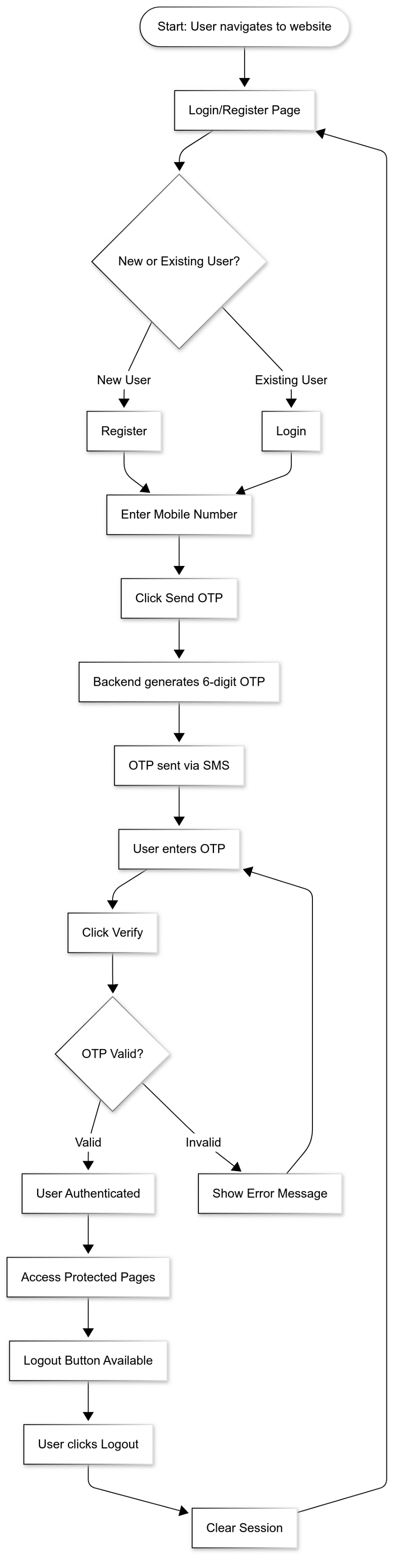
* **Entry Point:** All users land on a unified “Login / Register” page capturing phone numbers.
* **OTP Workflow:** Two‑step process with clear progress indicators, countdown timers, and resend options.
* **Error States:** Friendly messages for invalid numbers, failed deliveries, and expired codes.
* **Accessibility:** Keyboard‐navigable forms, ARIA alerts for validation messages, and focus management throughout the flow.

**7. Environment & Configuration**

**Objective:** Centralize operational configuration and safeguard sensitive credentials.

* **Environment Variables:** Securely manage database connection strings, SMS gateway credentials, and session secrets outside source control.
* **Deployment Strategy:** Use CI/CD pipelines to inject environment variables and rotate secrets periodically.
* **Disaster Recovery:** Document rollback procedures and have secrets versioned in a secure vault.

**Conclusion:** This report serves as the blueprint for business and operational stakeholders to understand the requirements for an OTP-based authentication system. Once approved, I can translate these requirements into technical implementation tasks, ensuring alignment with security policies and user‑experience goals.

****