Overview

As a Lead Test Engineer, my approach to testing the ENSEK application will be comprehensive, risk-based, and focused on delivering maximum value to the client. I'll create a structured testing strategy that covers all critical aspects of the application while demonstrating leadership in test planning and execution.

Understanding the Application

First, I need to understand what the application does. Based on initial exploration on https://ensekautomationcandidatetest.azurewebsites.net/, it appears to be energy-related application with the following main sections:

- ✦ Home
- ✦ About
- ✦ Contact
- ✦ Register
- ✦ Login
- ✦ Energy (with gas and electricity options)
- ✦ Buy energy functionality

Testing Approach

1. Initial Exploration (15% of effort)

- **Goal**: Understand application functionality and identify testable components
- **Activities**:
  - Manual exploration of all visible features
  - Review of user flows (registration, login, energy purchase)
  - Documentation of all observed functionality
  - Identification of potential risk areas

2. Functional Testing (40% of effort)

- **Goal**: Verify all features work as expected
- **Test Areas**:
  - **User Registration**
    - Valid/invalid email formats
    - Password requirements
    - Duplicate registration
    - Success notification
  - **User Login**
    - Valid credentials
    - Invalid credentials
    - Password recovery
    - Session management
  - **Energy Purchase Flow**
    - Gas/electricity selection
    - Quantity validation
    - Calculation accuracy
    - Payment simulation
    - Order confirmation
  - **Navigation**
    - Link validation
    - Menu functionality
    - Breadcrumbs (if present)
    - Back button behaviour

3. Non-Functional Testing (25% of effort)

- **Performance Testing**
  - Page load times
  - Response times under load
  - Concurrent user handling
- **Usability Testing**
  - Intuitive navigation
  - Clear error messages
  - Mobile responsiveness (if applicable)
  - Accessibility checks

4. Automation Strategy (20% of effort)

- **Goal**: Implement reusable automated tests for regression suite
- **Approach**:
  - Prioritise high-value, repetitive tests for automation
  - Use Selenium WebDriver with C-sharp for UI tests
  - API testing with Postman/Talend
  - Implement CI/CD integration
  - Create maintainable, modular test framework

Risk-Based Prioritisation

I'll prioritise testing based on:

1. **Critical Business Functions**: Energy purchase flow, payment processing
2. **High-Risk Areas**: User authentication, data validation
3. **Frequently Used Features**: Navigation, energy selection
4. **Visible Components**: UI consistency, error handling