**Test Plan for ENSEK Candidate Test System**

**1. Introduction**

This test plan outlines the approach and strategies for testing the ENSEK Candidate Test System. The system is a web-based application that allows users to buy energy of different types, view company information, and contact the company. The test plan covers both functional and non-functional testing, with a focus on ensuring the application behaves as expected under various scenarios.

**2. Scope**

The scope of this test plan includes:

* **Functional Testing**: Verification of all functional requirements and user interactions.
* **Non-Functional Testing**: Performance, security, and usability testing.
* **Error Handling**: Validation of error messages and error conditions.
* **User Interface (UI) Testing**: Consistency and correctness of the UI elements.
* **Data Validation**: Ensuring data integrity and correctness in input and output.

**3. Test Objectives**

* Verify that the application functions as expected across all supported browsers and devices.
* Ensure that all links and buttons work correctly.
* Validate that the application handles both valid and invalid inputs gracefully.
* Confirm that the application displays the correct information and updates data accurately.
* Identify and document any defects or issues found during testing.

**4. Test Strategy**

**4.1. Test Approach**

The testing approach will be a combination of manual and automated testing. Initially, manual testing will be performed to explore the application and identify areas for automation. Automated tests will be developed using a suitable testing framework (C# with Selenium WebDriver) to ensure repeated and consistent testing of critical paths.

**4.2. Test Levels**

* **Unit Testing**: *Not applicable for this test plan but should be included.*
* **Integration Testing**: *Not applicable for this test plan but should be included.*
* **System Testing**: Focus on functional and non-functional testing of the entire system.
* **Acceptance Testing**: Validate that the system meets the business requirements.

**4.3. Test Types**

* **Functional Testing**: Testing of individual functionalities.
* **Regression Testing**: Ensuring that changes do not break existing functionality.
* **Smoke Testing**: Quick sanity checks to verify the application is stable.
* **Performance Testing**: Evaluating the application's performance under load.
* **Security Testing**: Identifying potential security vulnerabilities.
* **Usability Testing**: Assessing the user-friendliness of the application.

**5. Test Scenarios and Test Cases**

**5.1. Home Page Testing**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| TC001 | Verify Home Page Loads Correctly | - | 1. Open the application URL. | The Home Page should load successfully with all sections (Home, About, Contact). | High | Pass |
| TC002 | Verify About Section Link | - | 1. Click on the "About" link. | The About page should load with the company description and "Find out more" button. | High | Pass |
| TC003 | Verify Contact Section | - | 1. Click on the "Contact" link. | Contact details should be displayed | Medium | Pass |
| TC004 | Verify Register Section | - | 1. Click on the "Register" link. | Register details should be displayed, Email , Password, Confirm Password | High | Pass |
| TC005 | Verify Login Section | - | 1. Click on the "Login" link. | Login details should be displayed, Email , Password , Login button | High | Pass |

**5.2. Buy Energy Functionality**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC006** | Verify Buy Energy Page Loads | - | 1. Click on "Buy energy" button. | The Buy Energy page should load with the current time and available energy types. | High | Pass |
| **TC007** | Verify Time Display on Buy Energy Page | - | 1. Navigate to Buy Energy page. | The time displayed should be the current time | Medium | Pass |
| **TC008** | Verify Energy Type Availability | - | 1. Check the availability of each energy type. | Gas: m3 units, Nuclear: MW units , Electricity: kWh units, Oil: Litres units. | High | Pass |
| **TC009** | Buy Gas | - | 1. Enter 20 units for Gas.  2. Click "Buy" button. | A confirmation page should display the purchase details and update the remaining units. | High | Pass |
| **TC010** | Buy More | Successful buy | 1. Click “Buy More” | Return to Buy Energy Page with update time | High | Pass |
| **TC011** | Buy Electricity | - | 1. Enter 10 units for Electricity.  2. Click "Buy" button. | A confirmation page should display the purchase details and update the remaining units. | High | Pass |
| **TC012** | Buy Nuclear | - | 1. Enter 30 units for Nuclear.  2. Click "Buy" button. | A confirmation page should display the purchase details and update the remaining units. | High | Skip |
| **TC013** | Buy Oil | - | 1. Enter 5 units for Oil.  2. Click "Buy" button. | A confirmation page should display the purchase details and update the remaining units. | High | Pass |
| **TC014** | Attempt to Buy More Gas Units Than Available | Reset for base state | 1. Enter 400000 units for Gas.  2. Click "Buy" button. | An error message should be displayed indicating insufficient units. | High | Fail |
| **TC015** | Attempt to Buy More Nuclear Units Than Available | Reset for base state | 1. Enter 222 units for Nuclear.  2. Click "Buy" button. | An error message should be displayed indicating insufficient units. | High | Fail |
| **TC016** | Attempt to Buy More Electricity Units Than Available | Reset for base state | 1. Enter 3434 units for Electricity.  2. Click "Buy" button. | An error message should be displayed indicating insufficient units. | High | Fail |
| **TC017** | Attempt to Buy More Oil Units Than Available | Reset for base state | 1. Enter 21 units for Oil.  2. Click "Buy" button. | An error message should be displayed indicating insufficient units. | High | Fail |
| **TC018** | Attempt to Buy Negative Units | - Reset for base state | 1. Enter -10 units for Gas.  2. Click "Buy" button. | An error message should be displayed indicating invalid input. | Medium | Fail |
| **TC019** | Attempt to Buy Non-Numeric Units | - Reset for base state | 1. Enter "abc" units for Gas.  2. Click "Buy" button. | An error message should be displayed indicating invalid input. | Medium | Fail |
| **TC020** | Attempt to buy huge amount | Reset for base state | 1. Enter "9999999999999" units for Gas.  2. Click "Buy" button. | An error message should be displayed indicating insufficient units. | Medium | Fail |
| **TC021** | Attempt to Buy Numeric with special chracter | Reset | 1. Enter "88\*" units for Electricity.  2. Click "Buy" button. | An error message should be displayed indicating invalid input. | Medium | Fail |
| **TC022** | Verify Reset Button Functionality | - | 1. Click "Reset" button. | The available units should reset to their initial values. | High | Pass |

**5.3. Sell Energy Functionality**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC023** | Verify Sell Energy Page Display | - | 1. Click on "Sell energy" button. | Sell Landing Page should be displayed | Medium | Fail |

**5.4. About Us Section**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC024** | Verify About Us Page Content | - | 1. Click on "About us" button. | The About Us page should display the company description and "Find out more" button. | High | Pass |
| **TC025** | Verify "Find out more about us" Button Functionality | - | 1. Click on "Find out more about us" button. | The button should redirect to the live ENSEK homepage. | High | Pass |

**5.1. Register Page Testing**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC026** | Successful Registration | - | 1. Enter a valid email and password meeting all criteria. 2. Click "Register". | User is registered successfully and redirected to the appropriate page. | High | Fail |
| **TC027** | Invalid Email Format | - | 1. Enter an invalid email format. 2. Click "Register". | An error message is displayed indicating invalid email format. | High | Pass |
| **TC028** | Password Too Short | - | 1. Enter a password shorter than 6 characters. 2. Click "Register". | An error message is displayed indicating password is too short. | High | Pass |
| **TC029** | Password Without Uppercase Letter | - | 1. Enter a password without an uppercase letter. 2. Click "Register". | An error message is displayed indicating password lacks an uppercase letter. | High | Pass |
| **TC030** | Password Without Digit | - | 1. Enter a password without a digit. 2. Click "Register". | An error message is displayed indicating password lacks a digit. | High | Pass |
| **TC031** | Password Without Non-Alphanumeric | - | 1. Enter a password without a non-alphanumeric character. 2. Click "Register". | An error message is displayed indicating password lacks a non-alphanumeric character. | High | Pass |
| **TC032** | Mismatched Passwords | - | 1. Enter mismatched passwords. 2. Click "Register". | An error message is displayed indicating passwords do not match. | High | Pass |

**5.2. Login Page Testing**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC033** | Successful Login | User is registered | 1. Enter correct email and password. 2. Click "Log in". | User is logged in successfully and redirected to the homepage. | High | Fail |
| **TC034** | Incorrect Email | - | 1. Enter an incorrect email. 2. Click "Log in". | An error message is displayed indicating incorrect email or password. | High | Fail |
| **TC035** | Incorrect Password | User is registered | 1. Enter correct email and incorrect password. 2. Click "Log in". | An error message is displayed indicating incorrect email or password. | High | Fail |
| **TC036** | Unregistered Email | - | 1. Enter an unregistered email. 2. Click "Log in". | An error message is displayed indicating incorrect email or password. | High | Fail |
| **TC037** | Remember Me Functionality | User is registered | 1. Check "Remember me" and log in. 2. Log out and log in again. | User is automatically logged in without entering credentials again. | Medium | Fail |

**5.6. Security Testing**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**5.7. Performance Testing**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC038** | Verify Application Performance Under Load | - | 1. Simulate multiple users buying energy simultaneously. | The application should handle the load without significant performance degradation. | High | Pass |

**5.8. Usability Testing**

| **Test Case ID** | **Test Case Title** | **Preconditions** | **Test Steps** | **Expected Result** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| **TC039** | Verify Ease of Use for Users | - | 1. Perform the buy energy process from start to finish. | The process should be intuitive and straightforward for users. | Medium | Pass |
| **TC040** | Verify Responsiveness on Different Devices | - | 1. Access the application on different devices (desktop, tablet, mobile). | The application should be responsive and function correctly on all devices. | High | Pass |

**6. Reusable Components and Parameterization**

To optimize the testing process and reduce the number of end-to-end tests required, we will create reusable components and parameterize them. This approach allows us to cover multiple test scenarios with fewer tests by reusing common actions and checks across different test cases. This will also be the basis for test automation components.

**6.1. Reusable Components**

**6.1.1. Navigation Component (Example in Appendix)**

* **Description**: A reusable component that handles navigation between pages.
* **Parameters**:
  + page: The target page to navigate to (e.g., "Home", "About", "Contact", “Register”, “Login”).
* **Usage**:
  + **Navigate to About Page**: navigationComponent(page="About")
  + **Navigate to Contact Page**: navigationComponent(page="Contact")

**6.1.2. Login Component**

* **Description**: A reusable component that handles the login process.
* **Parameters**:
  + Login: true or false (false skips component)
  + email: The email address used for login.
  + password: The password used for login.
  + rememberMe: A boolean indicating whether the "Remember me" option is selected.
* **Usage**:
  + **Successful Login**: loginComponent(email="user@example.com", password="Password1!", rememberMe=true)
  + **Incorrect Password**: loginComponent(email="user@example.com", password="wrongpassword", rememberMe=false)
  + **Unregistered Email**: loginComponent(email="newuser@example.com", password="Password1!", rememberMe=false)

**6.1.2. Register Component (Example in Appendix)**

* **Description**: A reusable component that handles the register process.
* **Parameters**:
  + register: true or false (false skips component)
  + email: The email address used for login.
  + password: The password used for login.
  + confirm\_password: The password used for login.
* **Usage**:
  + **Successful Regsitration**: loginComponent(register =”true”,email="user@example.com", password="Password1!")

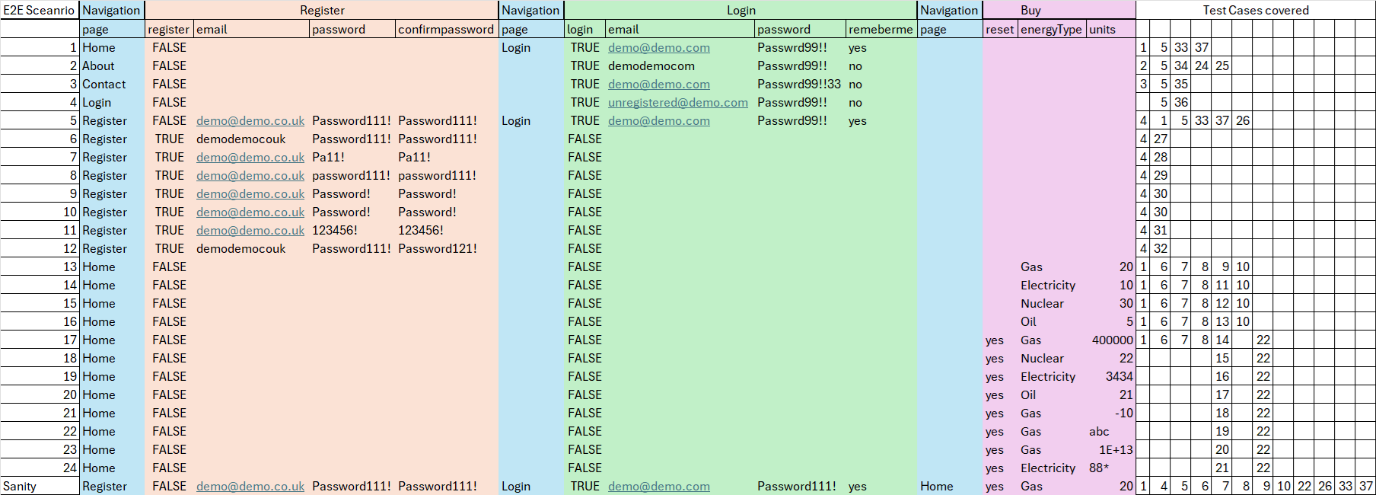
**6.1.3. Buy Energy Component**

* **Description**: A reusable component that handles the buy energy process.
* **Parameters**:
  + energyType: The type of energy to buy (e.g., "Gas", "Electricity", "Oil").
  + units: The number of units to buy.
* **Usage**:
  + **Buy Gas**: buyEnergyComponent(energyType="Gas", units=20)
  + **Buy Electricity**: buyEnergyComponent(energyType="Electricity", units=10)
  + **Buy Oil**: buyEnergyComponent(energyType="Oil", units=5)

**6.3. Test Execution Strategy**

* **Test Execution**: Each end-to-end scenario will be executed with the specified parameters to cover the associated test cases.
* **Test Reporting**: The test execution report will indicate which test cases were covered by each end-to-end scenario.

By implementing reusable components and parameterization, we can efficiently cover a wide range of test scenarios with fewer end-to-end tests, thereby optimizing the testing process.



(Attached in folder)

**7. Test Environment**

* **Browsers**: Google Chrome, Mozilla Firefox, Microsoft Edge
* **Operating Systems**: Windows 10, macOS, Linux
* **Devices**: Desktop, Tablet, Mobile

**8. Test Deliverables**

* **Test Plan Document**: This document.
* **Test Cases**: Detailed test cases with steps, expected results, and actual results.
* **Defect Reports**: Detailed reports for any defects found during testing.
* **Test Execution Reports**: Summary of test execution results.
* **Test Automation Scripts**
* **Regression Packs**

**9. Risk Management**

**9.1. Identified Risks**

* **Risk 1**: Incomplete or incorrect functionality in the Buy Energy feature.
  + **Mitigation**: Thorough testing of all buy energy scenarios.
* **Risk 2**: Security vulnerabilities in the application.
  + **Mitigation**: Conduct security testing and penetration testing.
* **Risk 3**: Performance issues under load.
  + **Mitigation**: Perform load testing to identify and resolve performance bottlenecks.

**Appendix**

A screenshot of a computer

Description automatically generated

***Figure 1: Example of the navigate component that can be used as reusable component.***

A screenshot of a computer

Description automatically generated

***Figure 2: Example of the Register component that can be used as reusable component.***