**High-Level Test Plan for Energy Visualization Platform**

**Objective:**  
The goal of the plan is to ensure that the energy visualization platform delivers accurate, real-time, and historical energy data to customers while providing valuable insights and recommendations to both customers and businesses.

**1**. **Test Coverage Areas**

* 1. **Functional Testing:**

1. Energy consumption data visualization for real-time data
2. Energy consumption data visualization for historical data
3. Comparing Energy usage between different time periods.
4. Checking alert systems for abnormal energy consumption/usage.
5. Test personalized recommendations for energy efficiency based on customer usage patterns.
6. Customizing the visual dashboard by the user.
7. Energy visualization feature on different devices
8. Energy usage comparison advisory between the user and the common average usage
   1. **Non-functional Testing:**

**Usability Testing**: Verifying that the user interface is easy to understand and easy to use for both customers and businesses.

**Security Testing**: Testing the security of the API service and the platform against OWASP standards. If cloud deployment is involved, using cloud security standards for the same. (Penetration testing is also needed)

**Accessibility Testing**: Validate that the platform complies with accessibility standards (WCAG) for a wide user base.

**Performance Testing (Load and performance)**: Test the platform’s responsiveness and load capacity for real-time data visualization.

**Performance testing the API Service**:

Testing the performance of the API service for load and responsiveness.

* 1. **Integration Testing:**

Testing the integration between the visual UI platform, Data API, Central Data Service, and Alerting System works well for all user stories.

* 1. **Compatibility Testing:**
* Ensure compatibility across multiple browsers (Chromium, Firefox etc) and devices (desktop, mobile, tablet).
  + 1. Browser Version specific
    2. OS version specific

2.**Test cases:**

Note: Assuming that the required test data for different customers is seeded to DB before testing these acceptance criteria.

**User Story 1:** Energy consumption data visualization for real-time data

AC 1: View real-time energy consumption data

**Given** that I am logged into the website

**When** I navigate to the dashboard

**Then** I should see a real-time energy consumption gauge in kWh values

**And** the data should refresh dynamically

**And** I should see the correct values

**Technique:**

* 1. Boundary values [ for verifying low, high or extreme consumption]
  2. API testing with a Data-driven approach

**User Story 2:** Energy consumption data visualization for historical data

AC 2: View historical energy consumption for selected periods

**Given** that I am logged into the website

**And** I navigate to the dashboard

**When** I select a specific period [last week, last month etc]

**Then** I should see a line chart displaying consumption for the selected period

**And** the data should be displayed correctly

**Technique:**

1. Equivalence Partitioning [ valid and Invalid inputs]

**Valid:** The user has data for the selected period

**Invalid**: The user has no data for the selected period

**User Story 3:** Comparing Energy usage between different time periods.

AC 3: View energy consumption usage between periods

**Given** that I am logged into the website

**And** I navigate to the dashboard

**When** I select two different time periods [last week and the previous week etc]

**Then** I should see a bar chart displaying consumption for the selected period

**And** the data should displayed correctly on daily basis

**Technique:**

1. Pairing values [ all possible combination : week , month,]
2. UI test covers the integration
3. API testing with a Data-driven approach

**User Story 4:** Checking alert systems for abnormal energy consumption/usage.

AC 4: Validate alert functionality for abnormal energy consumption

**Given** that I am logged into the website

**And** I have alert setup for specific threshold

**When** my energy consumption exceeds the threshold

**Then** I should receive a visual alert on the dashboard

**Technique:**

* 1. Boundary testing – setting different threshold
  2. UI test covers the integration

**User Story 5:** Test personalized recommendations for energy efficiency based on customer usage patterns.

AC 5: Verify personalized recommendations for energy efficiency

**Given** that I am logged into the website

**When** I navigated to the recommended section

**Then** I should see suggestions

**Technique:**

* 1. Exploratory testing – verify the recommendation relevance
  2. Algorithm testing – validate the output generated by Machine Learning

1. **User Story 6:** Customizing the visual dashboard by the user.

AC 6: Verify customization options to add a widget

**Given** that I am logged into the website

**When** I navigated to the dashboard settings

**Then** I should widgets

**When** I add a widget

**Then** the changes should be reflected in the UI with the new widget

AC 7: Verify customization options to remove a widget

**Given** that I am logged into the website

**When** I navigated to the dashboard settings

**Then** I should widgets

**When** I remove a widget

**Then** the changes should be reflected in the UI without the widget

**Technique:**

* 1. Exploratory testing – to test widget integration
  2. UX design testing – how the page renders or appearance when customizing the wizards

**User Story 7:** Energy visualization feature on different devices

AC 8: Accessing the feature on different devices

**Given** that I am a mobile device or tablet

**When** I access the website

**Then** the energy visualization feature should provide a responsive layout that adapts to different screen sizes.

**Technique:**

* 1. Exploratory testing – chrome browser Dimension[Responsive]
  2. Cross browser/ device testing – SauceLabs or BrowserSatck

**User Story 8:** Energy usage comparison advisory between the user and the common average usage

AC 9: Comparing advisory between the user and the common average usage

**Given** that I am logged into the website

**And** I don’t have any threshold setup

**When** my energy consumption exceeds the default average usage

**Then** I should receive a visual alert on the dashboard

**Technique:**

* 1. Boundary testing – viewing dashboard with different usage data

3**.Missing Information**

1. How is user authentication going to be handled?
2. Any Roles involved [Basic, Bronze, Silver etc] that affect the access to data?
3. How often consumption data will be updated in the DB?
4. What will happen If there is a DB outage or the service is down?
5. Alert Threshold limits are not specified and it makes it hard to do Boundary testing
6. Need more information towards Machine Learning algorithms about how the recommendations are generated

4**. Recommendation to enhance user experience**

**1.** Showing 2-3 minutes Videos to demo How to use the feature

2. Tooltip explaining how to do a comparison or how to add/delete a wizard

3. Enabling the user to choose the graph type [bar charts to line charts]

4. Letting user download the usage data in an Excel sheet to do offline analysis

5. Selling to the user if they can switch to a better energy plan which would save them money depending on usage patterns

6.

5**. Questions to Ask**

1. How many minutes once the system fetch data from DB to display the consumption dynamically

**2.** Are there any metrics available [ time-to-load the data] to do performance testing

3. How does the Machine learning Algorithm work? Can we test the logic?

4. Do we show pre-cached data in case the API is down

5. How to create an account for new customer to be onboarded ?