Business Requirements Document (BRD)

Project Title:

Highest Value Calculator

Document Purpose:

This document outlines the business requirements for a web-based functionality that allows a user to input three numeric values and returns the highest value among them, displayed on the screen.

---

1. **Project Overview**

The objective of this project is to develop a feature where a user enters three numeric values into three separate textboxes, and upon submitting, the highest value is calculated and displayed on the screen. The system should ensure that the input values are valid numbers. This functionality will enhance user experience by providing a simple and intuitive way to compare multiple values and see the largest one.

---

2. **Business Objectives**

The primary goal of this project is to create a responsive and user-friendly interface where:

1. Users can easily enter three numeric values.

2. The system identifies and displays the highest of the three values.

3. Basic validation checks ensure that the input values are numeric.

---

3. **Scope**

The scope of this project includes:

1. Frontend interface for value input.

2. Functionality to validate input data.

3. Logic to determine the highest value.

4. Display of the highest value.

5. Error handling for invalid or empty inputs.

Out of scope:

- Comparison of more than three values.

- Advanced mathematical operations (e.g., comparisons of decimals or percentages).

- Persistent storage of values (only in-memory operations are required).

---

4. **Functional Requirements**

4.1 **User Interface**

1. **Three Input Textboxes**:

   - The system must provide three textboxes where users can input numeric values.

   - Each textbox should be labeled accordingly (e.g., "First Value", "Second Value", and "Third Value").

2. **Submit Button**:

   - A button labeled "Find Highest" will be available for users to click after entering the values. Clicking this button triggers the calculation.

3. **Result Display Area**:

   - The highest of the three values will be displayed in a result area or alert box below the input fields once the user submits the values.

4. **Clear Button** (Optional):

   - An additional "Clear" button may be provided to reset all textboxes and the result area.

4.2 **Validation Requirements**

1. **Numeric Validation**:

   - The system must validate that all three textboxes contain valid numeric values.

   - If any input is invalid or left blank, the system should prompt the user with an error message like, "Please enter valid numbers in all fields."

2. **Handling Non-Numeric Values**:

   - If the user enters non-numeric values (e.g., letters, special characters), an error message must be shown.

3. **Empty Fields Validation**:

   - If any of the fields are empty, the system should prompt the user with an error message to fill in all fields.

4.3 **Business Logic Requirements**

1. **Highest Value Calculation**:

- The system must compare the three entered values and identify the highest.

- Once identified, the system should display the highest value on the screen.

4.4 **Error Handling**

1. If any of the fields contain invalid input or are left empty, the system must prevent the calculation and display a user-friendly error message like, "All fields must contain valid numbers."

---

5. **Non-Functional Requirements**

5.1 **Performance**

- The system must process input validation and return the highest value instantly upon user submission.

5.2 **Usability**

- The user interface should be intuitive, with clear labels for each input field and an easily identifiable submit button.

- The system should handle invalid inputs gracefully, guiding the user with appropriate error messages.

- The system must be compatible across modern web browsers (e.g., Chrome, Firefox, Safari, Edge).

5.3 **Security**

- The system should not store any user inputs beyond the current session (no persistence).

- Input validation should prevent the injection of malicious code.

5.4 **Accessibility**

- The interface must be accessible to users with disabilities, adhering to WCAG 2.1 AA standards. This includes proper labeling of textboxes and buttons for screen readers.

---

6. **Assumptions**

1. Users will only enter numeric values into the textboxes.

2. The system will be used on standard web browsers and does not require compatibility with older, outdated browsers.

3. No complex mathematical operations beyond comparison of the three values will be required.

---

7. **Dependencies**

1. **Front-End Development Tools**:

   - HTML5 for structuring the page.

   - CSS for styling the layout.

   - JavaScript for handling the business logic of the application.

2. **Browser Compatibility**:

   - The system should work across different browsers (Chrome, Firefox, Safari, and Edge).

---

8. **Acceptance Criteria**

1. **Input Validation**:

   - The system must validate that all textboxes contain numeric values before performing any calculation.

   - If invalid or empty input is detected, an appropriate error message must be shown.

2. **Calculation of Highest Value**:

   - The system must accurately calculate and display the highest value entered by the user.

3. **Result Display**:

   - The result must be displayed in a user-friendly format after the calculation is performed.

4. **Error Handling**:

   - If there are any invalid inputs, the system must prevent the calculation and alert the user to correct the input fields.

---

9. **Mockup/Prototype**

A simple prototype will include:

- **Three labeled input fields** for the user to enter numeric values.

- A **"Find Highest" button** to trigger the highest value calculation.

- A **result area** or alert to display the highest value after calculation.

- Optional: A **"Clear" button** to reset all inputs.

---

10. **Timeline**

Phase	Estimated Duration
Requirements Gathering	1 Week
Design (UI/UX and Mockups)	2 Weeks
Development	1 Week
Testing (Unit and User Testing)	1 Week
Deployment	1 Day

---

11. **Risks**

1. **Invalid Input Handling**: If validation logic is not implemented correctly, users may experience errors or incorrect results.

2. **Browser Incompatibility**: If not tested across different browsers, the system may not function as expected for some users.

---

12. **Conclusion**

This project aims to provide users with a simple interface to input three values and find the highest among them. The solution will be intuitive, responsive, and will guide users with appropriate error messages when invalid data is entered.

This concludes the Business Requirements Document for the Highest Value Calculator.