User Interface (UI)

1. **Employee Name Validation**
   - **Issue:** The UI allows users to enter numbers in the "First Name" and "Last Name" fields.
   - **Steps to Reproduce:**
     1. Click on "Add Employee" button.
     2. When form opens
     3. Enter numeric values in the "First Name" and "Last Name" fields.
     4. Submit the form and observe that numeric values are accepted without validation errors.
   - **Recommendation:** Implement validation to restrict these fields to alphabetic characters only.
2. **Error Messaging on Invalid Inputs**
   - **Issue:** No error messages or modals are displayed when invalid inputs are submitted.
   - **Steps to Reproduce:**
     1. Click on "Add Employee" button.
     2. Enter invalid data in required fields (e.g., leave mandatory fields empty or use incorrect formats).
     3. Attempt to submit the form and observe that no error messages appear.
   - **Recommendation:** Add real-time validation with error modals to inform users about invalid inputs.
3. **Dashboard Records Duplication**
   - **Issue:** The Dashboard permits adding more than 26 records for the same user.
   - **Steps to Reproduce:**
     1. Create or select an employee in the Dashboard.

        2. Add more than 26 records for that employee.
        3. Observe that the system allows duplicate entries beyond the specified limit.
    - **Recommendation:** Implement a restriction to prevent excessive duplicate records.

4. **Annual Benefits Cost Summation Discrepancy (Employee)**
    - **Issue:** The annual benefits cost for an employee sums to $999.96 instead of $1,000.
    - **Steps to Reproduce:**
        1. Select an employee with benefits.
        2. Calculate the total annual benefits cost and observe the $999.96 result instead of $1,000.
    - **Recommendation:** Adjust rounding and summing algorithms for accuracy.
5. **Annual Benefits Cost Summation Discrepancy (Dependent)**
    - **Issue:** The annual benefits cost for a dependent sums to $499.98 instead of $500.
    - **Steps to Reproduce:**
        1. Add a dependent for an employee.
        2. Calculate the total annual benefits cost for the dependent and observe it totals $499.98 instead of $500.
    - **Recommendation:** Adjust rounding or calculations as needed.

Application Programming Interface (API)

1. **Employee Name Validation**
    - **Issue:** The API allows numbers and special characters in "First Name" and "Last Name."
    - **Steps to Reproduce:**

        1. Send a POST request to add an employee with numbers or special characters in "First Name" and "Last Name."
        2. Verify the API accepts the input without validation errors.
    - **Recommendation:** Enforce strict alphabetic validation in these fields.

2. **Adding Employee - Dependents Requirement**
    - **Issue:** Dependents are not required when adding an employee.
    - **Steps to Reproduce:**
        1. Send a POST request to add an employee without specifying dependents.
        2. Observe that the employee is created despite the lack of dependents.
    - **Recommendation:** Make dependents a required field for new employee entries.
3. **Updating Employee - Dependents Requirement**
    - **Issue:** Dependents are not required in the PUT request to update an employee.
    - **Steps to Reproduce:**
        1. Send a PUT request to update an employee without dependents.
        2. Observe that the update is processed despite missing dependents.
    - **Recommendation:** Make dependents a required field in update requests for consistency.
4. **PUT Request - Resource Creation on Non-Existence**
    - **Issue:** The system creates a new entry if a PUT request is made for a non-existent resource.
    - **Steps to Reproduce:**
        1. Send a PUT request with data for an employee that does not exist.
        2. Observe that a new employee is created.
    - **Recommendation:** Enforce PUT for updates only and use POST for creation.
5. **Resource Re-creation with Zeroed Salary on Deletion**

- **Issue:** Updating a deleted employee with a PUT request creates a new entry with salary and gross pay set to zero.
- **Steps to Reproduce:**
    1. Delete an employee record.
    2. Send a PUT request to update this deleted employee.
    3. Verify that the re-created entry has salary and gross pay set to zero.
- **Recommendation:** Ensure retained values for recreated resources unless specifically altered.

6. **Resource Re-creation with Altered Parameters**
    - **Issue:** A deleted resource can be recreated with altered parameters, increasing fields like salary.
    - **Steps to Reproduce:**
        1. Delete an employee record.
        2. Send a PUT request to recreate the employee with a higher salary.
        3. Confirm that the recreated employee reflects the updated salary and gross pay.
    - **Recommendation:** Apply validation to prevent unauthorized modifications.
7. **ID Recycling and Resource Duplication with PUT**
    - **Issue:** Repeated PUT requests with a previously deleted ID create multiple resources.
    - **Steps to Reproduce:**
        1. Delete an employee record.
        2. Send a PUT request with the deleted ID to recreate the resource multiple times.
        3. Observe the system allowing creation of duplicate records.
    - **Recommendation:** Handle soft-deleted records separately to prevent duplicate entries.
8. **Inverted First and Last Names Across API Methods**

- **Issue:** First and last names are reversed in all REST API methods.
- **Steps to Reproduce:**
    1. Send any API request involving employee data (e.g., GET, POST).
    2. Observe the order of first and last names in the response.
- **Recommendation:** Correct name ordering across all methods for consistency with UI conventions.