**AI Usage Report**

I used an AI assistant to help me design a **helper method** that checks whether a JTextField is empty. The AI guided me on:

1. Writing a reusable method isFieldEmpty().
2. Passing both the text field and a label/field name to show which field is missing.
3. Integrating the method into the save button so all fields are validated before saving

**//Added a Helper method to check if each field is empty and throws an error if it is**

private boolean isFieldEmpty(String text) {

return text == null || text.trim().isEmpty();

}

private boolean validateFields() {

// Check each field and show error message immediately when empty field is found

if (isFieldEmpty(tfName.getText())) {

JOptionPane.showMessageDialog(this, "Name field is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfDescription.getText())) {

JOptionPane.showMessageDialog(this, "Description field is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfAvailability.getText())) {

JOptionPane.showMessageDialog(this, "Availability field is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfPrice.getText())) {

JOptionPane.showMessageDialog(this, "Price field is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfStrtName.getText())) {

JOptionPane.showMessageDialog(this, "Manufacture Street Name is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfUnitNum.getText())) {

JOptionPane.showMessageDialog(this, "Manufacture Unit Number is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfManCity.getText())) {

JOptionPane.showMessageDialog(this, "Manufacture City is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfManZipcode.getText())) {

JOptionPane.showMessageDialog(this, "Manufacture Zip Code is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfShpStrtName.getText())) {

JOptionPane.showMessageDialog(this, "Shipping Street Name is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfShpUnitNum.getText())) {

JOptionPane.showMessageDialog(this, "Shipping Unit Number is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfShpCity.getText())) {

JOptionPane.showMessageDialog(this, "Shipping City is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

if (isFieldEmpty(tfShpZipCode.getText())) {

JOptionPane.showMessageDialog(this, "Shipping Zip Code is required!", "Validation Error", JOptionPane.ERROR\_MESSAGE);

return false;

}

return true; // All fields are valid

}

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {

**//Added this section to the button save method**

// Validate fields first - if validation fails, stop here

if (!validateFields()) {

return; // Don't proceed with saving if validation fails

}

// If validation passes, proceeds with existing save logic

Conclusion

The AI helped me understand the benefit of abstraction by writing a single helper method that can be applied to multiple fields