1) Functions to find the address of a column by name:

```
Function Find_Col(sheet_name As String, Col_Name As String) As Range

    Sheets(sheet_name).Select
    Dim xRgUni As Range
    Dim xFirstAddress As String
    Dim xStr As String
    On Error Resume Next
    xStr = Col_Name
    Set Find_Col = Range("A1:DDD1").Find(xStr, , xlValues, xlWhole, , , True)
    If Not Find_Col Is Nothing Then
        xFirstAddress = Find_Col.Address
        Do
            Set Find_Col = Range("A1:DDD1").FindNext(Find_Col)
            If xRgUni Is Nothing Then
                Set xRgUni = Find_Col
            Else
                Set xRgUni = Application.Union(xRgUni, Find_Col)
            End If
        Loop While (Not Find_Col Is Nothing) And (Find_Col.Address <> xFirstAddress)
    End If

End Function
```

2) Function that will return the column letter of a column:

```
Function Alpha_Column(Cell_Add As Range) As String
Dim No_of_Rows As Integer
Dim No_of_Cols As Integer
Dim Num_Column As Integer
No_of_Rows = Cell_Add.Rows.count
No_of_Cols = Cell_Add.Columns.count
If ((No_of_Rows <> 1) Or (No_of_Cols <> 1)) Then
    Alpha_Column = ""
    Exit Function
End If
 Num_Column = Cell_Add.Column
If Num_Column < 26 Then
    Alpha_Column = Chr(64 + Num_Column)
```

```
Else

    Alpha_Column = Chr(Int(Num_Column / 26) + 64) & Chr((Num_Column Mod 26) +
64)
End If
End Function
```

3) Function for copy and paste columns (useful for filling in data for template):

```
Function Copy_Col_Pay(sheet_name As String, temp As String)

'Paste Name
  Sheets(sheet_name).Select
  Range("A2").Select
  Range(Selection, Selection.End(xlDown)).Select
  Selection.Copy

  Sheets(temp).Select
  Range("A2").Select
  ActiveSheet.Paste

End Function
```

4) Function for checking if cell content include duplicate words

```
Function IsDuplicates(rng As Range) As String
Dim StringtoAnalyze As Variant
Dim i As Integer
Dim j As Integer
Const minWordLen As Integer = 4
StringtoAnalyze = Split(UCase(rng.Value), " ")
For i = UBound(StringtoAnalyze) To 0 Step -1
If Len(StringtoAnalyze(i)) < minWordLen Then GoTo SkipA
For j = 0 To i - 1
If StringtoAnalyze(j) = StringtoAnalyze(i) Then
IsDuplicates = "TRUE"
GoTo SkipB
End If
```

```
Next j
SkipA:
Next i
IsDuplicates = "FALSE"
SkipB:
End Function
```

5) Function that return words that are duplicate in a range:

```
Function DuplicatedWords(rng As Range, Optional CaseSensitive As Boolean) As
Variant
  Dim X As Long, WordCount As Long, List As String, Duplicates As Variant, Words() As
String
  List = WorksheetFunction.Trim(Replace(Join(WorksheetFunction.Transpose(rng)),
Chr(160), " "))
  Words = Split(List)
  For X = 0 To UBound(Words)
    If CaseSensitive Then
      If UBound(Split("  " & List & "  ", " " & Words(X) & " ")) > 1 Then
        Duplicates = Duplicates & Words(X) & " "
        List = Replace(List, Words(X), "", 1, -1, vbBinaryCompare)
      End If
    Else
      If UBound(Split(" " & UCase(List) & " ", " " & UCase(Words(X)) & " ")) > 1 Then
        Duplicates = Duplicates & StrConv(Words(X), vbProperCase) & " "
        List = Replace(List, Words(X), "", 1, -1, vbTextCompare)
      End If
    End If
  Next
  Duplicates = WorksheetFunction.Trim(Duplicates)
  Words = Split(Duplicates)
  If Application.Caller.count > UBound(Words) Then
    Duplicates = Duplicates & space(Application.Caller.count - UBound(Words))
  End If
  DuplicatedWords = WorksheetFunction.Transpose(Split(Duplicates))
End Function
```

6) Sub for formatting data in a certain format:

```
Sub Intial_setup()
```

```
ThisWorkbook.Save
    Cells.Select

    Application.ScreenUpdating = False
    Application.Calculation = xlCalculationManual
    Application.EnableEvents = False

    If ActiveSheet.AutoFilterMode Then ActiveSheet.AutoFilterMode = False

    With Selection.Font
        .Name = "Calibri"
        .Size = 11
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .TintAndShade = 0
        .ThemeFont = xlThemeFontMinor
        .Color = vbBlack
        .Bold = False
        .Italic = False
    End With


    With Selection
        .WrapText = False
        .VerticalAlignment = xlCenter
        .HorizontalAlignment = xlCenter
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With

   'Change sheet fill color based on name
    With Selection
        Dim ws1 As Worksheet
        Dim Rng5 As Range
        Set Rng5 = ActiveSheet.UsedRange
        With ActiveSheet
```

```
        For Each ws1 In ActiveWorkbook.Worksheets
            If ActiveSheet.Name = "Applicants" Then
                Rng5.Interior.ColorIndex = 37
                ActiveSheet.Tab.ColorIndex = 37
            ElseIf ActiveSheet.Name = "New Hires" Then
                Rng5.Interior.ColorIndex = 40
                ActiveSheet.Tab.ColorIndex = 40
             Else
                Rng5.Interior.ColorIndex = 0
            End If
        Next ws1
    End With

End With

'Add border
With Selection
    Dim Rng7 As Range
    Set Rng7 = ActiveSheet.UsedRange
    For Each Cell In Rng7
    Cell.BorderAround _
        LineStyle:=xlContinuous, _
        Weight:=xlThin
    Next Cell
End With
'
' Change the first row color
With Selection
    Dim Rng6 As Range
    Set Rng6 = ActiveSheet.UsedRange.Rows(1)
    Rng6.Interior.ColorIndex = 27
    Rng6.Font.Bold = True
End With
'
'Unhide,Unfreeze
Cells.EntireRow.Hidden = False
Cells.EntireColumn.Hidden = False
ActiveWindow.FreezePanes = False
'
' Auto fit
Selection.EntireColumn.AutoFit
Selection.EntireRow.AutoFit

 Application.EnableEvents = True
```

```
    Application.Calculation = xlCalculationAutomatic
    Application.ScreenUpdating = True

End Sub
```

7) Sub for remove blank row and columns:

```
Sub remove_blank()

   ThisWorkbook.Save

   Application.ScreenUpdating = False
   Application.Calculation = xlCalculationManual
   Application.EnableEvents = False

   If ActiveSheet.AutoFilterMode Then ActiveSheet.AutoFilterMode = False

    'Delete blank row
      Dim LastRowIndex As Integer
      Dim RowIndex As Integer
      Dim UsedRng As Range

      On Error Resume Next
      Set UsedRng = ActiveSheet.UsedRange
      LastRowIndex = UsedRng.Row - 1 + UsedRng.Rows.count

      For RowIndex = LastRowIndex To 1 Step -1
         If Application.CountA(Rows(RowIndex)) = 0 Then
            Rows(RowIndex).Delete
         End If
      Next RowIndex
   '
   '


   'Delete Blank column
      Dim iCntr
      Dim RnG2 As Range
      Set RnG2 = ActiveSheet.UsedRange
         For iCntr = RnG2.Column + RnG2.Columns.count - 1 To RnG2.Column Step -1
            If Application.WorksheetFunction.CountA(Columns(iCntr)) = 0 Then
Columns(iCntr).EntireColumn.Delete
         Next
```

```
    Application.EnableEvents = True
    Application.Calculation = xlCalculationAutomatic
    Application.ScreenUpdating = True

End Sub
```