Abstract:
In Small and Medium Enterprises (SMEs), all the reporting tasks are carried out in Microsoft Excel. The employees spend all of their time working on the reports and in the case of an error in the report, a tremendous amount of their time is incurred on the detection of that error. At one of the leather footwear companies in Lahore, Pakistan, report automation was carried out using visual basic for Application (VBA) in Microsoft Excel. The purpose of automation was to increase the reporting efficiency and minimize the chance of error. The authors automated the generation of production plan papers, which used to take 3.11 minutes to be made per plan paper. 3.11 minutes were required just for a single order of only one footwear article. This research provides the framework for the automation of manual reporting in Microsoft Excel. This automation was conducted by using VBA in Microsoft Excel. In the VBA code, the loops and conditional statements were used to program the manual activities to be performed in the report. Initially, the manual method was demonstrated in detail then way of report automation was the focus of discussion. The comparison of both methods was conducted in terms of time utilization. The manual method encompassed a series of activities whereas; the automated template included the buttons with few clicks. A time study of report-making by manual and automated method was conducted which indicated that the automated method was 1.36 minutes faster than the manual method. This research contributes to the provision of a detailed framework, with the help of which any manual work in Microsoft Excel can be automated. It was also indicated by this research that SMEs who cannot afford the implementation of Enterprise Resource Planning (ERP) software, have the option of VBA in Microsoft Excel in which they can enhance their reporting efficiency and office employees’ productivity.

Keywords: production; plan papers; visual basic for applications; macros.

1. Introduction
In the context of data computation, processing, and recovery, information technology is extremely important. A management information system is the main device used by organizations to store, organize and use data for decision-making (Karim, 2011). Producing useful information from collected sets of data is the goal of data management and organization. For administrative reporting, spreadsheets are frequently utilized in small and medium-sized enterprises, planning, covering costing, etc. spreadsheets can be used for a variety of tasks including data entry, analysis, visualizing, and storing. All of the above-mentioned tasks need the usage of numerous spreadsheet applications (Broman et al., 2017). The practitioner’s experiences have demonstrated that it is difficult to produce a reliable spreadsheet (Dunn, 2009). The spreadsheet is the most used application by office employees across the globe these days (Fisher et al., 2002). Spreadsheets are used by both individuals and corporations for a variety of tasks, such as doing rapid calculations (Abraham et al., 2008). Large businesses need a complex information system to complete these duties; nonetheless, a comparable customized application can be made using VBA and Microsoft (MS) Excel if the dataset is not too big. Users of Microsoft Excel applications, use VBA in the form of ‘macros’ that are used for the automation of routine office work (Perry, 2012). Report automation through VBA in Microsoft Excel is a cost-effective option for organizations, with the help of which a tremendous amount of reporting time can be saved along with high accuracy (Blayney & Sun, 2019). The macros can be executed with just a click of a button after they are properly programmed and individuals can program the user-defined functions (USDFs) in a customized way by using VBA; in this way,
the calculations and reporting tasks in Microsoft Excel can be made easier and less time-consuming (Abraham et al., 2008). VBA is quite distinctive from Microsoft Excel formulas and is coded in the visual basic editor (VBE) of the Microsoft Office applications (Microsoft Word, Excel, and Access) (Abraham et al., 2008). Balson (2012) provided instructions on how to use user-defined spreadsheet functions that can shortly be accessible in Microsoft Excel. It is quite possible to program the relevant USDFs in the VBE module with a dynamic input range (Balson, 2012). Botchkarev (2015) reported that Microsoft Excel VBA is a very good platform to conduct Monte Carlo simulations (MCSs) after investigating its applicability (Botchkarev, 2015). The scope and budget of building work, comprising data mining and Production Planning and Control (PPC), can be carried out using Microsoft Excel (Ajinkya et al., 2017). Raza and Gulwani used VBA in Excel to find suspicious computing opinions and fill in the blanks for misplaced points (Raza & Gulwani, 2017). In the field of industrial reporting, such as price, budgets, scheduling, etc., very few research articles on report automation have been published. It is illustrated in this research how conditional decision-making can automatically be conducted with VBA.

The implementation of enterprise resource planning (ERP), material resource planning (MRP), and various other software that are used for processing the production data; the employees take out reports from them and they require quite less time and are accurately made. SMEs that cannot afford the implementation of enterprise resource planning (ERP) software have the option of VBA in Microsoft Excel by which they can enhance their reporting efficiency and office employees’ productivity. This research contributes to the provision of a detailed framework, with the help of which manual work (which is conducted manually in Microsoft Excel) in Microsoft Excel can be automated. This research also provides the VBA code in appendices (that was used during the automation of the present report) for the reference of industry professionals who want to carry out report automation in their routine work.

2. Literature review

The format of the report was kept the same based on the demand of the case company’s planning department, and the report no longer contained any manual operation, as it had been completely removed by the usage of visual basic for applications. In this context, the study on automation/optimization utilizing VBA is given. The Air Pollutant Index (API) and Water Quality Index (WQI) were calculated by the research of Zainal Abidin et al. (2015) using VBA in Microsoft Excel. The software was configured to calculate the indices directly. The necessary formulas were converted into code. Additionally, each index description was programmed so that the measured index value's specifics could be displayed automatically alongside it (Abidin et al., 2015). Ahmadi and colleagues (2018) carried out a study to implement the CTR Dairy model through VBA in Excel. The CTR Dairy model is a dynamic simulation model utilized to predict the milk production and profits of grazing lactating dairy cows. The model takes various factors into account such as nutrient absorption and urinal digestion, and considers intermittent feeding schedules to make its predictions. The conversion of the CTR dairy model into Excel VBA transformed its input into output due to the unavailability of its clients and the termination of the SMART software. This research made the model accessible to a wide range of dairy nutrition consultants, academicians, dairy farmers, and extension advisers (Ahmadi et al., 2018). The demand management system is improved at logistic centre of civil aviation authority (CAA) Karachi by using VBA which reduced 50% of working time (Chaudhry et al., 2021).

Junior et al. (2011) used Excel VBA to create a new program for post-processing RELAP5 findings. It is said to be a helpful tool for accelerating output data analysis (Belchior Junior et al., 2011). In 2009, Rushit Hila developed an application that automatically identifies outliers and arranges data using VBA in Excel. The application includes automated steps for cleaning and verifying field data before importing it into MS Access, which serves as the database (Hila, 2009). A novel Cirujano and Zhu (2013) developed a novel technique using VBA Excel for automatically producing personnel planning reports. The engineers' responsibilities and availability for various projects were gathered using the new methodology. The collected data was assembled, examined, and arranged. Plans for the engineers might be developed in this respect by retrieving the engineer's information on various projects. The technique was tested at a consulting company with more than 100 workers. There was a suggestion that personnel planning reports may be prepared automatically, saving a significant amount of time as well as cash (Cirujano & Zhu, 2013). Sato and Yokoyama (2001) used Excel VBA to create an application that allows users to move picture data from a data set to a spreadsheet by clicking on a Web icon (Sato & Yokoyama, 2001). Lessa et al. (2016) utilized Visual Basic Application (VBA) in Excel to automate a mathematical model for the computation of packing and logic programs. The program automatically data graphics to display to the user instead of the packages (Lessa et al., 2016). In 2014, H Evensen used Excel VBA to implement the communication with instruments in Excel, marking the first instance of such implementation (Evensen, 2014). An automatic report production system was created by Donald E. Blattner and Valrico, Florida (2007) using VBA in MS Projects. The created system allowed users to choose, sort the report, and format the filter using the on-screen assistance dialogue box (Blattner & Valrico, 2007). To apply mapping rules, Wettlaufer (2010) created VBA macros in Excel. One macro was coded for each report. The macros wrote the anticipated values in another spreadsheet, referred to as the expected values spreadsheet. The data from the patient follow-up sessions are then sent to the Merlin net server for processing, and a report package in a WinRAR file is created as a result (Wettlaufer, 2010).

Footwear sector is proliferating worldwide but in the context of electronic automation, it lags behind (Arain et al., 2020; Kalwar & Khan, 2020b). The recent automation based applied case studies of footwear industry Pakistan include procurement report (Kalwar & Khan, 2020a), routine report making (Kalwar et al., 2020), purchase order (Kalwar & Khan, 2020c), acquisition report (Kalwar & Khan, 2020a), planning report (Kalwar et al., 2021), Supplier Price Evaluation Report (Khan,
Kalwar, Malik, et al., 2021), material delivery time analysis (Khan, Kalwar, & Chaudhry, 2021), product mix & profit maximization (Kalwar, Khan, Shahzad, et al., 2022), and order costing analysis (Kalwar, Shahzad, et al., 2022). More recently, the automation was implemented at material cost comparative analysis (MCCA) at footwear industry which resulted in 100% accuracy and 72.20% time reduction in 58.51 minutes (Kalwar et al., 2023).

VBA was used by Norton and Tiwari (2013) to create code that would help engineering students grasp the analysis of innovative freezing technology (Norton & Tiwari, 2013). Bartoszewicz and Wdowicz (2019) put forward a novel approach for analyzing and automating the production process. Their approach involved integrating the production planning module of SAP enterprise resource planning (ERP) with an Excel worksheet and VBA to enable a faster and more flexible method of data migration and analysis. This new approach significantly improved the speed of the complex analytical report, reducing operation time from 2 hours to just 5 minutes (Bartoszewicz & Wdowicz, 2019).

In 2017, Harahap and Azmi conducted a study aimed at developing a VBA Excel application capable of creating a small rainwater conveyance system using the MSMA 2 logical method. The researchers aimed to produce a new tool that could efficiently implement the MSMA 2 approach (Harahap & Azmi, 2017). In 2017 Yan and Wan created an application using Excel VBA for automatic computation and creation of the gearbox line’s bill of materials (BOM). The layout and implementation of the template significantly increased accuracy and efficiency and decreased the number of mistakes made when creating the entire steel BOM (Yan & Wan, 2017).

Optimization of the operations is considered among the top priorities in manufacturing sector (Bukhsh et al., 2021; Iftikhar et al., 2021; Iftikhar, Khan, et al., 2022; Iftikhar, Kumar, et al., 2022; Jaleel et al., 2021; Khan et al., 2021; Rajput et al., 2020). The process industries should utilise the potential of modern production techniques and optimization tools to improve their overall productivity (Haseeb et al., 2023). Operational excellence is much needed to remain competitive for organizations (Kalwar, Khan, Phul, et al., 2022; Kumar et al., 2020). The process industries are required to transform themselves with the dynamic business environment (Kumar et al., 2022). Lean Manufacturing (LM) philosophy is getting popularity for optimizing operations in printing, pharmaceutical, automobile, textile and manufacturing industries of Pakistan (Khan, Soomro, et al., 2020; Khan, Memon, et al., 2020; Khan, 2018; Khan, Marri, et al., 2020; Khan, Khatri, et al., 2020). The globally practiced concepts of sustainable supply chain (Junejo, Qureshi, et al., 2023), ERP (Kalwar, Khan, Phul, et al., 2022), TQM (Memon et al., 2022) and Circular Economy (Sheikh et al., 2022) still have the great potential in Pakistani Industry.

3. Research methodology

The employee at the case company, who was supposed to make the report manually, provided the information on the procedure and approaches utilized to create the reports as per the manual method. After the authors understood how data is fetched and how it flows across the workbook; the macros were then programmed to automate the manual tasks. The macros were programmed in the default visual basic editor of Microsoft Excel and the code included the nested for loops and conditional statements to fetch and process the data. Moreover, the loops were used to transfer the data from one worksheet to another, and conditional statements were used to verify the data to be transferred in the worksheets. The programmed macros were executed through the command buttons put on the user form, and it was set to pop up on ‘Ctrl + Q’. The plan papers could be generated by a few clicks in the automated template. The flowcharts representing the sequence of macros’ execution were made in Microsoft Word, and the snipping tool (a default snipping app) was used to take snaps of the screens that are included in the discussion of results. A stopwatch was used to record the length of each activity performed during the making of the production plan paper report as per the manual and automated methods. All the observations were entered into MS Excel to compute the average time to complete a particular activity. The comparison of the times taken by both methods was conducted to reflect the difference/improvement brought by the automated method. Using a stopwatch was the only way the authors had to use for the measurement of the time consumption in the generation of plan papers via both methods.

This report automation could also be carried out using C# but it will increase the report preparation time because the employees will have to put all the data manually and that would take a tremendous amount of time. The second possibility to carry the automation of plan papers generation was to embed the module for it in Microsoft Dynamics AX, which is presently being used by the company; if the company does this, it would incur a greater cost to the company. These are the reasons, the authors chose VBA programming to automate the generation of production plan papers and the good thing is that the employees are already aware of how to use Microsoft Excel.

4. Results

4.1. Existing method of making papers

Plan papers were created manually in Microsoft Excel as part of the manual process that was in place at the time.

4.1.1. Time required to make report manually

A stopwatch was then used to record the times of the many tasks that had to be completed to create a plan paper. Microsoft Excel was used to calculate the average time to complete each task based on the ten observations for each activity. Table 1 lists all the activities and their corresponding needed times.

Figure 1 shows the average time needed to accomplish each task; the longest time was spent downloading the data (x1) from the ERP, which is shown in Figure 1.

4.2. Suggested method

VB for Application in Microsoft Excel was used to build the template and served as the foundation for the suggested approach.
4.2.1. Required worksheets and their function in automated template

The APP template needs four worksheets to function as intended. Each worksheet is connected to the minor others, and while a report is being created, data from numerous sheets is fetched at various levels. When downloading bills of material (BOMs) from Microsoft AX Dynamics, copying them into the worksheet for bills of materials (BOMs), and entering the article number and colour as shown in Figure 2, is showing examples of how to use the worksheet.

The “freeze articles” worksheet asks for information on the articles (such as article number, colour, sale order (SO) number, and week number) for which the plan papers are to be created (see Figure 3).

![Figure 2: Worksheet of a bill of materials (BOMs).](image-url)
Templates for doing computations are arranged as shown in Figure 5. Figure 4 illustrates how the sale order detail worksheet offers the article specifics while the plan documents include information about the number of pairs and sale order number.

**4.2.2. Data Fetching Across Various Worksheets**

Figure 6 shows how several worksheets are connected in terms of automatic data access. The details about articles, such as colour, production week, pairings, and SO number are first transferred into each production plan paper using BOMs, and these details are taken from the worksheet titled "frozen articles".

For data processing for the APP template, the “Assortment” worksheet is left empty. The assortment worksheet's template compares the values in the worksheet for “sell order details” to the number of pairs that must be scheduled that were entered in the worksheet for “freeze goods” for the APP template to function in the context of producing plan papers, the data is therefore gathered from several worksheets.

**4.2.3. The function of Automated Template**

The user form was created with command buttons as shown in Figure 7 to execute the macros that were built.

---

**Figure 3:** Details of frozen articles as discussed in the freeze meeting.

**Figure 4:** Worksheet of Sale order detail as pasted from sale order detail report.

**Figure 5:** Worksheet of assortment; this worksheet is required for the automated processing of data taken from the sale order detail worksheet.

**Figure 6:** Data fetching among worksheets in the automated template.
to automate the generation of plan documents. The user form appears when the user taps Ctrl+Q.

Figure 7: Designed form to execute macros in the automated plan papers template.

1.1.1. Execution Sequence of Command Buttons

The numbers of macros programmed into the backs of five of the command buttons, as well as the order in which they are executed, are shown in Figure 8.

4.2.3.1. Initialize

Eleven macros are located behind the first command button, “initialize” and they are executed in the order shown in Figure 9. Appendix 1 contains the macro code that powers the “initialize” command button.

The output in the automatically inserted worksheets for plan papers is received as shown in Figure 10 when all of the macros in Figure 9 are executed.

4.2.3.2. Generate

Nine macros are located behind the second command button, “create” and they are run sequentially as seen in Figure 11. Appendix 2 contains the macro code that powers the “initialize” command button.

The output in the automatically inserted worksheets for plan papers is received as shown in Figure 12 when all of the macros in Figure 11 are executed.

4.2.3.3. Convert to PDF

Back of the command button, there is only one macro labelled “convert to pdf”, and the VBA code for it is provided in Appendix 3. The created plan papers, as seen in Figure 13, contain a pdf document that is the macro’s output.

4.2.3.4. Delete Sheets

The automated inserting of sheets by the template for the development of plan papers is deleted as the fourth phase of command buttons and macros. Appendix 4 contains the code for the “remove sheets” command button.

Figure 8: Command buttons and macros at their back end.

Figure 9: Macros and their execution sequence when the command button (initialize is clicked).
Automation of production plan generating workbook at leather footwear company of Lahore Pakistan by using VBA in Microsoft Excel

4.2.4. Time Required to Make plan papers with Automated Template

The time required to make the plan papers by APP template is given in Table 2. The table is enlisted with the execution time of each command button. In summing up the time of all the activities, it totals 81.875 seconds or 1.36 minutes.

4.3. Comparison of old and suggested methods

The manual production plan paper for one article used to take 186.843 seconds or 3.11 minutes from the employee’s time. Since there were 11 plan papers; thus 34.21 minutes were supposed to be spent on making plan papers. Eleven plan documents were created using the APP template in 1.36 minutes. The comparison showed that using this template helped the employee save 96.02% of their time.

5. Discussion

Every small and medium-sized company hires personnel to produce Excel reports on a monthly, quarterly, daily, and annual basis. Employee time is wasted on manual Excel operations, and manual labour has a higher likelihood of mistakes. In this way, software providers have maintained the automation option in office programs like Excel, PowerPoint, and Word as well as Microsoft Projects, etc. As a result, Microsoft created technologies like VSTO, VBA, ActiveX, and a lot more to meet user demands (Ding et al., 2017; Porter & Stretcher, 2012). There are various software applications available on the Windows platform that incorporate VBA technology (Kuka & Karamani, 2011; Norton & Tiwari, 2013; Harahap & Azmi, 2017). Currently, businesses encourage their staff to learn how to use Excel and Visual Basic for Applications. (Chatvichiencha, 2015) VBA technology pertains to simplifying and automating complex and repetitive tasks through development and customization within the integrated development environment (IDE) of office applications (Ding et al., 2017; Evensen, 2014; Minto,
The routine tasks carried out by existing productivity at work programs are automated using it (Ding et al., 2017; Chatvichiencha, 2015). The goal of the current study was to completely automate all manual processes, including creating the report. VBA uses user forms to collect information from users (Evensen, 2014). The author of the current work employed a combo box on the form to collect input from the user (a report he wishes to work on). In order to carry out the duties automatically, user forms may additionally have codes and activities at their back (Evensen, 2014; Harahap & Azmi, 2017; Kalwar & Khan, 2020c). To automate the creation of plan papers, a user form was developed with command buttons that execute specific tasks through macros with a single click. Bartoszewicz and Wdowicz (2019) utilized VBA technology to redesign and implement a faster and more flexible process for data migration and analysis. This allowed for a complex analytical report to be generated in a shorter amount of time, reducing the operation time from two hours to five minutes (Bartoszewicz & Wdowicz, 2019). Using VBA in Excel, Cirujano and Zhu (2013) automated the manpower resource planning report. Prior to automation, an experienced reporter would take thirty hours to complete the report manually. However, after the implementation of VBA technology, the report can now be generated in just ten minutes, resulting in a 994% reduction in the time required (Cirujano & Zhu, 2013). Seventy-five percent (75%) of the time that staff would have otherwise spent manually creating the procurement report was saved by automation in footwear operations (Kalwar & Khan, 2020c). The process of generating the order costing report was significantly faster after automation compared to the manual technique, with a time reduction of 8592%. Yan and Wan (2017) created an application using Excel VBA for automatic computation and creation of the gearbox line’s bill of materials (BOM). The design and use of the template significantly increased accuracy and efficiency and decreased the number of mistakes made when creating the entire steel BOM (Yan & Wan, 2017). With the use of VBA, Abidin et al. (2013) automatically computed WQI and API. The program offered a useful method for calculating WQI and API, while also reducing computation time as well as error after automation (Abidin et al., 2015). Dynamic HRM policies are the need of the hour for industries now and improved HR Architecture (Khan et al., 2021), Green HR (Rahoo, Khan, et al., 2020), Emotional Intelligence (Arafin, Rahoo, et al., 2021), modern HR practices (Khan, 2003), Organizational Culture (Rahoo, Channar, et al., 2020; Rahoo, Hasnain, et al., 2021) improve the job satisfaction. ICT implementation can improve the operations performance & customer satisfaction (Kalhoro et al., 2019; Kalhoro, Abro, Shaikh, et al., 2018; Arshad et al., 2018; Kalhoro, Abro, Kalhoro, et al., 2018; Memom et al., 2020) but the main hurdle in it is the lack of fund (Nagar et al., 2018) and footwear industry has no exception. The post Covid-19 situation and E-Commerce changed the purchase intentions of local customers (Junejo, Kumar, et al., 2023; Rahoo, Ramejo, et al., 2021; Baladi et al., 2021).

6. Conclusion

When technology or the science of art is embedded in a machine, it repeatedly performs that operation with the desired accuracy; whereas, if humans are supposed to perform the same task, the chance of human error is certain. In this regard, the work that was repeatedly done by the employees and they used to spend a lot of time fixing the mistakes made by them. In this regard, the repetitive manual work in Microsoft was automated and left for a computer to perform. The computer did that work with 100% accuracy and after the automation, the production plan papers were made in 96.02% time as compared to the manual method. It means 960.2% time of an employee was saved that was supposed to be spent only on the generation of production plan papers. Additionally, programming and output verification for the template took a week. An employee had the work which needed human attention and in this way, the other creative ideas were focused by the help of which everyday processes at the organization could be developed further. After the automation of this report, whose time was saved came up with greater creativity and innovative thinking.

7. Future Implications

The template’s user lacked the expertise required to fix any coding mistakes. The researcher provided VBA instruction, but due to his hectic schedule, no one was able to take the course. The entire company neither acted simultaneously nor had the necessary VBA expertise.

8. Limitations

This template was adequate for the generation of production plan papers up to 200, if the number exceeds 200, the template will take more time to generate the output. Since the company owns an ERP, it can hire the developers to get the module developed for the generation of production plan papers but it will incur a greater cost to the company. This is the reason, the author chose VBA for this automation.

9. Acknowledgement

I would like to express our appreciation to the co-authors for their supporting and encouraging advice. I want to express my gratitude to the company’s staff for their genuine collaboration throughout this endeavour.

10. Conflict Of Interest

The authors of the current study had no conflicts of interest.

References


Perry, K.M. (2012). The Call Center Scheduling Problem using Spreadsheet Optimization and VBA. Virginia Commonwealth University.


APPENDICES

APPENDIX 1

MACRO 1

Sub DragArticlesAndColor()
rowscount = Worksheets("BOMS"),Cells(Rows.count, 4).End(xlUp).row
For i = 1 To rowscount
If Worksheets("BOMS"),Cells(i, 1).Value = " " Then
With Worksheets("BOMS")
.Cells(i, 1).Value = Cells(i - 1, 1)
.Cells(i, 2).Value = Cells(i - 1, 2)
End With
End If
Next
End Sub

MACRO 2

Sub PasteValuesInCol4()
Dim rowcount As Long
rowcount = Worksheets("Freeze Articles"),Cells(Rows.count, 1),End(xlUp).row
Range("A1:F1" & rowcount),Sort
key1:=Range("B1:B1" & rowcount), order1:=xlAscending, Header:=xNo
For i = 1 To rowcount
If Worksheets("Freeze Articles"),Cells(i, 1).Value = Worksheets("Freeze Articles"),Cells(i, 1).Value & "," Then
Worksheets("Freeze Articles"),Cells(i, 4).Value = 0
Else
Worksheets("Freeze Articles"),Cells(i, 4).Value = Cells(i, 4).Value + Cells(i, 1).Value
End If
Next
End Sub

MACRO 3

Sub InsertAndNameSheets()
On Error Resume Next
Dim rowcount As Long
Dim SheetName As String
Rowcount = Worksheets("Freeze Articles"),Cells(Rows.count, 1),End(xlUp).row
For i = 1 To rowcount
SheetName = Worksheets("Freeze Articles"),Cells(i, 1).Value & "," & Worksheets("Freeze Articles"),Cells(i, 2).Value & "," & Worksheets("Freeze Articles"),Cells(i, 4).Value & ","
Worksheets.Add_Name = SheetName
MsgBox(SheetName)
Next
End Sub

MACRO 4

Sub MergePlanCells()
Dim rowscount As Integer
MsgBox(rowscount, text = ",
rowcount = Worksheets("Freeze Articles"),Cells(Rows.count, 1),End(xlUp).row
MsgBox(rowscount + 
For i = 1 To rowscount
SheetName = Worksheets("Freeze Articles"),Cells(i, 1).Value & "," & Worksheets("Freeze Articles"),Cells(i, 2).Value & "," & Worksheets("Freeze Articles"),Cells(i, 4).Value & ","
MsgBox(SheetName)
Next
End Sub

Worksheets(SheetName).Select
With ActiveSheet
.Range("B1"), .Value = "Production plan"
End Sub

Automation of production plan generating workbook at leather footwear company of Lahore Pakistan
by using VBA in Microsoft Excel

rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.count, 1).End(xlUp).row
MsgBox (rowcountfreeze)
For i = 1 To rowcountfreeze
    SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & "" & Worksheets("Freeze Articles").Cells(i, 2).Value & "" & Worksheets("Freeze Articles").Cells(i, 4).Value & ",",
    article = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 1)
color = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 2)
    quantity = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 3)
    row = 6
    *MsgBox (SheetName)
    For j = 2 To rowcount
cell1 = ThisWorkbook.Worksheets("Assortment").Cells(j, 10).Value
cell2 = ThisWorkbook.Worksheets("Assortment").Cells(j, 11).Value
    If Worksheets("Assortment").Cells(j, 8) = article And Worksheets("Assortment").Cells(j, 9) = color
        quantity = quantity
        Worksheets("Assortment").Select
        Worksheets("Assortment").Range(1, 1, 1).Select
        ActiveSheet.Cells(Rows.count, 1).End(xlUp).row = 1
        If ActiveSheet.Cells(j, 2).Value = "Prod" Then
            ActiveSheet.Range(1, 10, Cells(1, colcount)).Copy
            .Cells(j + 1, 2).PasteSpecial xlPasteValues .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).Font.Bold = True
            .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).HorizontalAlignment = xlCenter
            .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).VerticalAlignment = xlCenter
        End If
Next
End Sub

Macro 4
Sub PasteSizeAndQty()
    Dim rowcountfreeze As Integer
    Dim rowcount As Integer
    Dim colcount As Integer
    rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.count, 1).End(xlUp).row
    For i = 1 To rowcountfreeze
        SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & "" & Worksheets("Freeze Articles").Cells(i, 2).Value & "" & Worksheets("Freeze Articles").Cells(i, 4).Value & ",
        article = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 1)
color = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 2)
    quantity = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 3)
        row = 6
        *MsgBox (SheetName)
        For j = 2 To rowcount
cell1 = ThisWorkbook.Worksheets("Assortment").Cells(j, 10).Value
cell2 = ThisWorkbook.Worksheets("Assortment").Cells(j, 11).Value
        If Worksheets("Assortment").Cells(j, 8) = article And Worksheets("Assortment").Cells(j, 9) = color
            quantity = quantity
            Worksheets("Assortment").Select
            Worksheets("Assortment").Range(1, 1, 1).Select
            ActiveSheet.Cells(Rows.count, 1).End(xlUp).row = 1
            If ActiveSheet.Cells(j, 2).Value = "Size" Then
                ActiveSheet.Range(Cells(2, 10), Cells(3, colcount)).Copy
                .Cells(j + 1, 2).PasteSpecial xlPasteValues .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).Font.Bold = True
                .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).VerticalAlignment = xlCenter
            End If
        End If
Next
End Sub

Macro 3
Sub PasteProd()
    Dim rowcountfreeze As Integer
    Dim rowcount As Integer
    Dim colcount As Integer
    rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.count, 1).End(xlUp).row
    For i = 1 To rowcountfreeze
        SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & "" & Worksheets("Freeze Articles").Cells(i, 2).Value & "" & Worksheets("Freeze Articles").Cells(i, 4).Value & ",
        article = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 1)
color = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 2)
    quantity = ThisWorkbook.Worksheets("Freeze Articles").Cells(i, 3)
        row = 6
        *MsgBox (SheetName)
        For j = 2 To rowcount
cell1 = ThisWorkbook.Worksheets("Assortment").Cells(j, 10).Value
cell2 = ThisWorkbook.Worksheets("Assortment").Cells(j, 11).Value
        If Worksheets("Assortment").Cells(j, 8) = article And Worksheets("Assortment").Cells(j, 9) = color
            quantity = quantity
            Worksheets("Assortment").Select
            Worksheets("Assortment").Range(1, 1, 1).Select
            ActiveSheet.Cells(Rows.count, 1).End(xlUp).row = 1
            If ActiveSheet.Cells(j, 2).Value = "Size" Then
                ActiveSheet.Range(Cells(2, 10), Cells(3, colcount)).Copy
                .Cells(j + 1, 2).PasteSpecial xlPasteValues .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).Font.Bold = True
                .Range(Cells(j + 1, 2), Cells(j + 1, 
colcount)).VerticalAlignment = xlCenter
            End If
        End If
Next
End Sub
Automation of production plan generating workbook at leather footwear company of Lahore Pakistan by using VBA in Microsoft Excel

Macro 6

Sub ClearExtraProd()
Dim rowcountfreeze As Integer
Dim rowcount As Integer
Dim colcount As Integer
rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.Count, 1).End(xlUp).Row
For i = 1 To rowcountfreeze
    SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & "+" & Worksheets("Freeze Articles").Cells(i, 2).Value & "+" & Worksheets("Freeze Articles").Cells(i, 4).Value & "+"
    WorkSheets(SheetName).Select
    With ActiveSheet
        rowcount = ActiveSheet.Cells(Rows.Count, 1).End(xlUp).Row
        For j = 1 To rowcount
            If ActiveSheet.Cells(j, 2).Value = "Prod" Then
                For k = 1 To colcount
                    If .Cells(j + k, 1).Value = "=" Then
                        Range(Cells(j, k), Cells(j + 1, k)).ClearContents
                    End If
                Next
                Range(Cells(j, k), Cells(j + 1, k)).ClearFormats
                Exit For
            End If
            .Cells(j + k, 1).Value = "="
            .Cells(j + k, 1).Value = "="
        Next
    End With
Next
End Sub

Macro 8

Sub PutBaskets()
Dim rowcountfreeze As Integer
Dim rowcount As Integer
Dim colcount As Integer
Dim qty As Integer
Dim NumOfBaskets As Integer
Dim row As Integer
rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.Count, 1).End(xlUp).Row
For i = 1 To rowcountfreeze
    SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & "+" & Worksheets("Freeze Articles").Cells(i, 2).Value & "+" & Worksheets("Freeze Articles").Cells(i, 4).Value & "+"
    WorkSheets(SheetName).Select
    With ActiveSheet
        rowcount = ActiveSheet.Cells(Rows.Count, 1).End(xlUp).Row
        For j = 1 To rowcount
            If ActiveSheet.Cells(j, 1).Value = "Pairs" Then
                For k = 2 To colcount
                    qty = .Cells(j, k).Value
                    NumOfBaskets = qty / 5
                    If qty > NumOfBaskets + 5 Then
                        NumOfBaskets = NumOfBaskets + 1
                    End If
                Next
            End If
        Next
    End With
Next
End Sub
Sub PutPlanNoNo()
    Dim rowcountfreeze As Integer
    Dim Comp Filpath As String
    Dim logo As Image
    Dim ArticleName As String
    Dim mypic As Picture
    Dim i As Integer
    rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.Count, 1).End(xlUp).Row
    For i = 1 To rowcountfreeze
        SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & ":" &Worksheets("Freeze Articles").Cells(i, 2).Value & " &Worksheets("Freeze Articles").Cells(i, 4).Value & ""
        ArticleName = "sp" & Worksheets(SheetName).Select
        With ActiveSheet
            .Insert Plan Number
            .Cells(4, 7).Value = "Plan# " &Worksheets("Freeze Articles").Cells(i, 5).Value
            .Cells(5, 8).Value = "Req. Qty"
            .Insert Logo
            .Range("A1:A3").Select
            .Comp Filpath = "Y:\Pictures\Art Pics\" & ArticleName & ".PNG"
        End With
        logo.Height = "65"
        logo.Width = "90"
        .Insert Articile Picture
        .ArticleName = "Worksheets("Freeze Articles").Cells(i, 1).Value
        .Range("F1:F3").Select
        .Comp Filpath = "Y:\Pictures\Art Pics\" & ArticleName & ".PNG"
        Set mypic = ActiveSheet.Pictures.Insert(Comp Filpath)
        ActiveSheet.Range("F1:F3").mypic.Height = "32"
        mypic.Width = "60.59"
    Next
End Sub

Macro 10
Sub PutReqQty()
    Application.ScreenUpdating = False
    Application.DisplayAlerts = False
    On Error Resume Next
    Dim rowcountfreeze As Integer
    Dim Pairs As Double
    Dim ReqQty As Double
    rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.Count, 1).End(xlUp).Row
    For i = 1 To rowcountfreeze
        SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & ":" &Worksheets("Freeze Articles").Cells(i, 2).Value & " &Worksheets("Freeze Articles").Cells(i, 4).Value & ""
        Pairs = Worksheets("Freeze Articles").Cells(i, 3).Value
        Worksheets(SheetName).Select
        With ActiveSheet.run = 6
            row = row + 1
        Loop
        End With
    Next
End Sub

Macro 11
Sub PutReferenceNumber()
    Dim rowcountfreeze As Integer
    rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.Count, 1).End(xlUp).Row
    For i = 1 To rowcountfreeze
        SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & ":" &Worksheets("Freeze Articles").Cells(i, 2).Value & " &Worksheets("Freeze Articles").Cells(i, 4).Value & ""
        RefNum = Worksheets("Freeze Articles").Cells(i, 6).Value
        Worksheets(SheetName).Select
        With ActiveSheet.run = Cells(Rows.Count, 2).End(xlUp).Row
            Do Until Cells(1, 1).Value = "Customer
                .Cells(j, 1).Value = "Col Ref#"
                .Range(Cells(j, 2), Cells(j, colcount)).MergeCells = True
                .Range(Cells(j, 2), Cells(j, colcount)).HorizontalAlignment = xlCenter.Range(Cells(j, 2), Cells(j, colcount)).VerticalAlignment = xlCenter.Range(Cells(j, 2), Cells(j, colcount)).Borders.LineStyle = xlContinuous.Range(Cells(j, 2), Cells(j, colcount)).Borders.Weight = xlThin.End If
            Next
        End With
    Next
End Sub

APPENDIX 3
Sub ExportASPDF()
Automation of production plan generating workbook at leather footwear company of Lahore Pakistan by using VBA in Microsoft Excel

APPENDIX 4

Sub DeleteArticleWorksheets()
Dim rowcountfreeze As Integer
rowcountfreeze = Worksheets("Freeze Articles").Cells(Rows.Count, 1).End(xlUp).Row
For i = 1 To rowcountfreeze
    SheetName = Worksheets("Freeze Articles").Cells(i, 1).Value & "" & 
    Worksheets("Freeze Articles").Cells(i, 2).Value & "" & 
    Worksheets("Freeze Articles").Cells(i, 4).Value & ""
    Worksheets(SheetName).Select
    With ActiveSheet
        .Delete
    End With
Next
End Sub