Order-Level-Python-Code

```
import pandas as pd
file_name = "invoice_Annexure_180796_09042025_1744207214805.xlsx"
df_summary = pd.read_excel(file_name, sheet_name='Summary')
summary_data = {
    "Brand": df_summary.iloc[3, 1],
    "Res-ID": df_summary.iloc[6, 1].replace("Rest. ID - ", "") if pd.notna(df_summary.iloc[6, 1]) else None,
    "Payout Period": df_summary.iloc[10, 2],
    "File Name": file_name
}
df_order = pd.read_excel(file_name, sheet_name='Order Level', header=2)
df_order.columns = df_order.columns.str.replace('\n', ' ').str.strip()

desired_columns = [
    'Order ID', 'Parent Order ID', 'Order Date', 'Order Status', 'Order Category',
    'Order Payment Type', 'Cancelled By?', 'Coupon type applied by customer',
    'Item Total', 'Packaging Charges',
    'Restaurant Discounts (Promo, Freebies, Flat Off, etc.)',
    'Swiggy One  Exclusive Offer Discount',
    'Restaurant Discount Share [3a+3b]',
    'Net Bill Value (before taxes) [1+2-3]', 'GST Collected',
    'Total Customer Paid [4+5]', 'Commission charged on', 'Service Fees %',
    'Commission', 'Long Distance Charges', 'Discount on Long Distance Fee',
    'Pocket Hero Fees', 'Swiggy One Fees', 'Payment Collection Charges',
    'Restaurant Cancellation Charges', 'Call Center Charges',
    'Delivery Fee sponsored by Restaurant (w/o tax)', 'Bolt Fees',
    'GST on Service Fee @18%', 'Total Swiggy Fees [6+7+8-9+10+11+12+13+14+15+16]',
    'Customer Cancellations', 'Customer Complaints',
    'Complaint & Cancellation Charges [17+18]', 'GST Deduction', 'TCS', 'TDS',
    'Total Taxes [19+20+21]', 'Net Payout for Order (after taxes) [A-B-C-D]',
    'Long Distance Order', 'Last Mile (in km)', 'MFR Accurate?', 'MFR Pressed?',
    'Coupon Code Sourced', 'Discount Campaign ID', 'Replicated Order',
    'Base order ID', 'Cancellation time', 'Pick Up Status', 'Swiggy One Customer?',
    'Pocket Hero Order?'
]
df_cleaned = df_order[desired_columns].copy()
meta_df = pd.DataFrame([summary_data] * len(df_cleaned))
final_df = pd.concat([meta_df.reset_index(drop=True), df_cleaned.reset_index(drop=True)], axis=1)
file1=pd.DataFrame(final_df)
file1
```

Note:

Here in the **file_name** we will specify the path for all the files and we will extract the information of all details what required. In the similar way we have to write the code by changing the files and getting the information After that we should merge all the files together.

The Python Code

```
all_files = [file1, file2, file3, file4, file5, file6, file7, file8, file9, file10, file11, file12]
merged_df = pd.concat(all_files, ignore_index=True)
merged_df
```

we are writing the output to the excel file

```
final_output=merged_df.to_excel("Order-Level.xlsx")
final_output
```