```
!pip install reportlab

import sys
from reportlab.lib import colors
from reportlab.lib.pagesizes import A4
from reportlab.platypus import SimpleDocTemplate, Table, TableStyle, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.units import inch



def get_invoice_data():
    """Defines the static data for the default invoice."""
    return {
        "invoice_number": "INV-2025-001",
        "date": "23/11/2025",
        "due_date": "30/11/2025",

        "my_company": {
            "name": "Acme Tech Solutions",
            "address": "123 Innovation Drive, Tech City",
            "email": "billing@acmetech.com",
            "phone": "+91 98765 43210",
        },

        "client": {
            "name": "Global Marketing Corp",
            "contact": "Jane Doe",
            "address": "456 Commerce Tower, Business Hub",
            "gstin": "22ABCDE1234F1Z5"
        },

        "line_items": [
            {"description": "Python Backend Development (2 weeks)", "quantity": 1, "rate": 25000.00},
            {"description": "Database Optimization and Testing", "quantity": 1, "rate": 15000.00},
            {"description": "Monthly Hosting & Maintenance Fee", "quantity": 1, "rate": 5000.00},
            {"description": "Custom UI/UX Design Consultation (10 hours)", "quantity": 10, "rate": 800.00},
        ],

        "notes": "Thank you for your business. Please ensure payment is made by the due date."
    }

def calculate_totals(line_items, tax_rate=0.18):
    """Calculates the subtotal, tax, and grand total."""
    subtotal = sum(item['quantity'] * item['rate'] for item in line_items)
    tax_amount = subtotal * tax_rate
    grand_total = subtotal + tax_amount

    return {
        "subtotal": subtotal,
        "tax_rate": f"{tax_rate*100:.0f}% GST",
        "tax_amount": tax_amount,
        "grand_total": grand_total,
    }

def format_currency(amount):
    """Formats a number as Indian Rupee (INR) currency."""
    return f"₹ {amount:,.2f}"

def create_pdf(canvas, data, totals):
    """
    Constructs the main elements of the invoice PDF.

    Args:
        canvas (list): The list of flowable elements for the PDF.
        data (dict): The invoice data including company, client, and items.
        totals (dict): Calculated totals.
    """
    styles = getSampleStyleSheet()


    header_data = [
        [
            Paragraph('<font size="24" color="#333333">INVOICE</font>', styles['Heading1']),
            Paragraph(f'<b>{data["my_company"]["name"]}</b><br/>{data["my_company"]["address"]}<br/>Email: {data[
        ]
    ]
    header_table = Table(header_data, colWidths=[3.5 * inch, 3 * inch])
    header_table.setStyle(TableStyle([
        ('ALIGN', (0,0), (-1,-1), 'LEFT'),
        ('VALIGN', (0,0), (-1,-1), 'TOP'),
        ('VALIGN', (0,0), (-1,-1), 'TOP'),
        ('BOTTOMPADDING', (0, 0), (-1, -1), 20),
    ]))
    canvas.append(header_table)

    canvas.append(Spacer(1, 0.25 * inch))


    detail_data = [
        ["Invoice Number:", data['invoice_number']],
        ["Invoice Date:", data['date']],
        ["Payment Due:", data['due_date']],
    ]
    detail_table = Table(detail_data, colWidths=[1.5 * inch, 1.5 * inch])
    detail_table.setStyle(TableStyle([
        ('GRID', (0, 0), (-1, -1), 0.5, colors.HexColor('#CCCCCC')),
        ('BACKGROUND', (0, 0), (0, -1), colors.HexColor('#F5F5F5')),
        ('FONTNAME', (0, 0), (-1, -1), 'Helvetica-Bold'),
        ('FONTSIZE', (0, 0), (-1, -1), 10),
        ('LEFTPADDING', (0, 0), (-1, -1), 6),
    ]))


    client_info = Paragraph(
        f'<b>Bill To:</b><br/>'
        f'<b>{data["client"]["name"]}</b><br/>'
        f'{data["client"]["contact"]}<br/>'
        f'{data["client"]["address"]}<br/>'
        f'GSTIN: {data["client"]["gstin"]}',
        styles['Normal']
    )


    billing_table_data = [
        [client_info, detail_table]
    ]
    billing_table = Table(billing_table_data, colWidths=[4 * inch, 3 * inch])
    billing_table.setStyle(TableStyle([
        ('VALIGN', (0, 0), (-1, -1), 'TOP'),
    ]))
    canvas.append(billing_table)

    canvas.append(Spacer(1, 0.5 * inch))




    table_data = [
        ['#', 'Description', 'Qty', 'Rate (₹)', 'Amount (₹)']
    ]


    for i, item in enumerate(data['line_items']):
        amount = item['quantity'] * item['rate']
        table_data.append([
            i + 1,
            Paragraph(item['description'], styles['Normal']),
            item['quantity'],
            format_currency(item['rate']),
            format_currency(amount)
        ])


    item_table = Table(table_data, colWidths=[0.3 * inch, 4.2 * inch, 0.7 * inch, 1.0 * inch, 1.0 * inch])


    item_table.setStyle(TableStyle([
        ('BACKGROUND', (0, 0), (-1, 0), colors.HexColor('#2A64D8')), # Header color
        ('TEXTCOLOR', (0, 0), (-1, 0), colors.white),
        ('FONTNAME', (0, 0), (-1, 0), 'Helvetica-Bold'),
        ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
        ('ALIGN', (2, 1), (-1, -1), 'RIGHT'), # Align Qty, Rate, Amount to the right
        ('GRID', (0, 0), (-1, -1), 1, colors.HexColor('#DDDDDD')),
        ('FONTSIZE', (0, 0), (-1, -1), 10),
        ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),
        ('ROWBACKGROUNDS', (0, 1), (-1, -1), [colors.HexColor('#FFFFFF'), colors.HexColor('#F5F5F5')]),
    ]))
    canvas.append(item_table)

    canvas.append(Spacer(1, 0.25 * inch))


    totals_data = [
        ['', 'Subtotal:', format_currency(totals['subtotal'])],
        ['', 'Subtotal:', format_currency(totals['subtotal'])],
        ['', totals['tax_rate'], format_currency(totals['tax_amount'])],
        ['', 'GRAND TOTAL:', format_currency(totals['grand_total'])],
    ]

    totals_table = Table(totals_data, colWidths=[5.2 * inch, 1.0 * inch, 1.0 * inch])

    totals_table.setStyle(TableStyle([
        ('ALIGN', (1, 0), (-1, -1), 'RIGHT'),
        ('GRID', (1, 0), (-1, -1), 1, colors.black),
        ('FONTNAME', (1, 2), (-1, 2), 'Helvetica-Bold'),
        ('FONTSIZE', (1, 2), (-1, 2), 12),
        ('BACKGROUND', (1, 2), (-1, 2), colors.HexColor('#C8DAF8')),
    ]))
    canvas.append(totals_table)

    canvas.append(Spacer(1, 0.5 * inch))


    canvas.append(Paragraph(f'**Notes:** {data["notes"]}', styles['Italic']))

    canvas.append(Spacer(1, 1.0 * inch))
    canvas.append(Paragraph('**Authorized Signature**', styles['Heading4']))


def generate_invoice(invoice_data, output_file_name):
    """Main function to run the invoice generation process."""
    data = invoice_data
    totals = calculate_totals(data['line_items'])

    output_filename = output_file_name

    try:
        print(f"Generating invoice: {output_filename}")

        doc = SimpleDocTemplate(output_filename, pagesize=A4,
                                topMargin=inch/2, bottomMargin=inch/2,
                                leftMargin=inch/2, rightMargin=inch/2)


        story = []


        create_pdf(story, data, totals)


        doc.build(story)

        print(f"Successfully generated invoice: {output_filename}")

    except Exception as e:
        print(f"An error occurred during PDF generation: {e}")
        sys.exit(1)

if __name__ == '__main__':
    default_invoice = get_invoice_data()
    generate_invoice(default_invoice, f"Invoice_{default_invoice['invoice_number'].replace('-', '_')}.pdf")
```

```
Collecting reportlab
  Downloading reportlab-4.4.5-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: pillow>=9.0.0 in /usr/local/lib/python3.12/dist-packages (from reportlab) (11.3.0)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.12/dist-packages (from reportlab) (3.
Downloading reportlab-4.4.5-py3-none-any.whl (2.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 19.2 MB/s eta 0:00:00
Installing collected packages: reportlab
Successfully installed reportlab-4.4.5
Generating invoice: Invoice_INV_2025_001.pdf
Successfully generated invoice: Invoice_INV_2025_001.pdf
```

```
new_invoice_data = {
    "invoice_number": "INV-2025-002",
    "date": "25/11/2025",
    "due_date": "05/12/2025",

    "my_company": {
        "name": "Acme Tech Solutions",
        "address": "123 Innovation Drive, Tech City",
        "email": "billing@acmetech.com",
        "phone": "+91 98765 43210",
    },

    "client": {
        "name": "Innovate Solutions Inc.",
        "contact": "John Smith",
        "address": "789 Creative Lane, Innovation Park",
        "gstin": "07ABCD1234E5Z6"
    },

    "line_items": [
        {"description": "Website Redesign Project", "quantity": 1, "rate": 35000.00},
        {"description": "Content Creation (5 articles)", "quantity": 5, "rate": 2000.00},
        {"description": "SEO Consultation (8 hours)", "quantity": 8, "rate": 1000.00},
    ],

    "notes": "Payment is expected within 10 days of the invoice date. Thank you!"
}

new_invoice_filename = f"Invoice_{new_invoice_data['invoice_number'].replace('-', '_')}.pdf"
generate_invoice(new_invoice_data, new_invoice_filename)

from google.colab import files
files.download(new_invoice_filename)
```

```
Successfully generated invoice: Invoice_INV_2025_002.pdf
```

```
Start coding or generate with AI.
```

```
from google.colab import files

files.download('Invoice_INV_2025_001.pdf')
```