NAME: ASHNA.   REGISTRATION NUMBER : 25BSA10138

1.PROBLEM STATEMENT:

Business frequently need to generate invoices for clients . The goal is to build a simple, efficient and customizable Python based invoice Generator that allows users to input customer details, item details and automatically generate professional Pdf invoice because many existing tools are paid or too complex for small scale users.

2.PROJECT OBJECTIVES:

1.Build end-to-end invoice generation system using python.

2.Automate the creation of professional PDF invoices

3.Provide a clear workflow with input validation.

4.Implement modular codebase with multiple functional modules.

5. Deliver proper documentation, diagrams, and GitHub structure.

3.FUNCTIONAL REQUIREMENTS

MODULE1 – User Input Module

Accept customer details, Accept invoice number and data, Collect multiple items like description quantity and price and validate inputs.

MODULE2- Module Processing

Calculate invoice grand total, apply formatting rules, calculate per-item total, store invoice data in CSV.

MODULE3 – PDF Generation Module

Use ReportLab

Generate a professional invoice layout

Save invoice as PDF, display confirmation

4.Non-Functional Requirements

Usability: Simple Command Line interface

Performance: Generate invoice PDF within 1 second

Maintainability: Modular code with separation of concerns

Scalability: Easy to add GUI or web interface later

5.SYSTEM ARCHITECTURE(TEXT DESCRIPTION)

INPUT DATA

Processing Module

(Calculations + Data)

PDF Generator Module

UML/ Design Diagrams

Use Case(Text Description)

Actors: User

Use cases

Enter Customer data

Enter item details

Generate invoice

Save invoice

6.CLASS DIAGRAM(Text Description)

Class Invoice:

-invoice_no

-customer_name

-items[]

+add_items()

+calculate_total()

Class Item:

-description

-quantity

-price

Class PDF Generator:

+ generate(invoice)

7.SEQUENCE DIAGRAM

User -> InputModule: Enter details

InputModule -> Invoice: Create invoice object

Invoice -> ProcessingModule: calculate totals

ProcessingModule -> PDFGenerator: request PDF

PDFGenerator -> FileSystem: save invoice.pdf

8. Folder Package Structure

Invoice_generator

    main.py

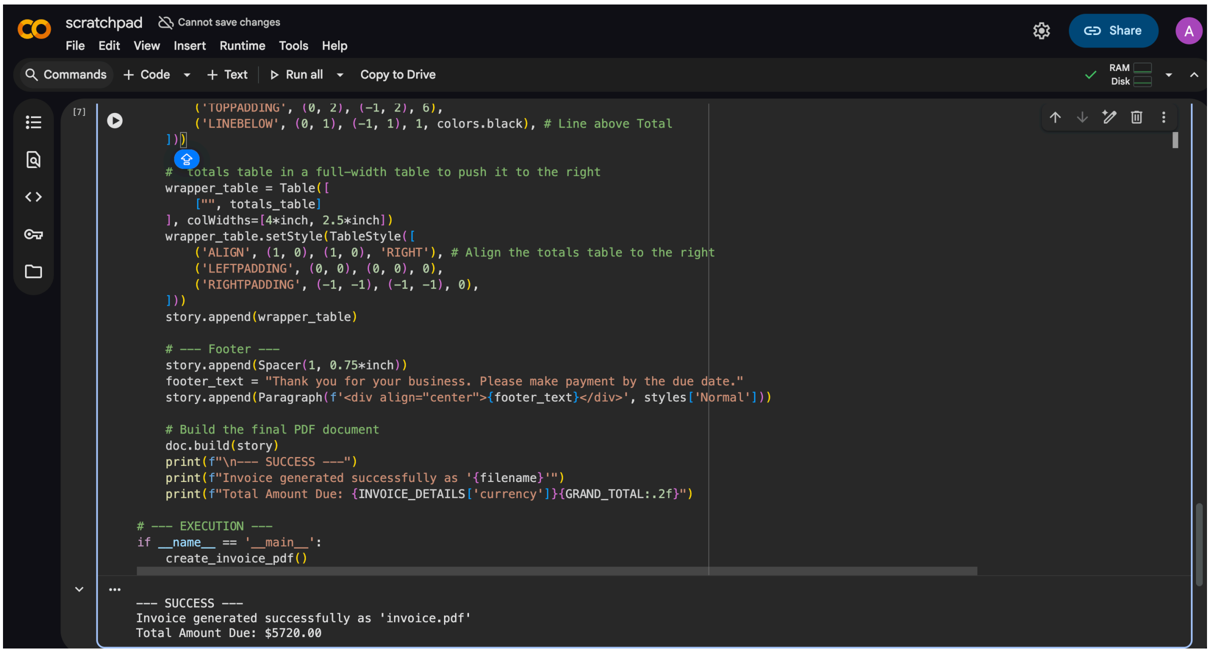
        Modules/

CODE:

```
rom reportlab.lib import colors
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Table, TableStyle, 
Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.units import inch

# --- 1. DATA DEFINITION ---
```

```python
INVOICE_DETAILS = {
    "id": "INV-2025-0042",
    "date": "November 20, 2025",
    "due_date": "December 20, 2025",
    "tax_rate": 0.10, # 10%
    "currency": "$"
}

COMPANY_INFO = {
    "name": "BetaTech Solutions",
    "address": "456 Python Lane, Silicon Valley, CA 94043",
    "phone": "(555) 555-1234",
    "email": "billing@alphatech.com"
}

CLIENT_INFO = {
    "name": "AlphaCorp Inc.",
    "address": "101 Code Street, Tech City, NY 10001",
    "contact": "Jane Doe"
}

# (Description, Quantity, Unit Price)
ITEMS = [
    ("Frontend Design Consultation (Hours)", 40, 75.00),
    ("Database Configuration & Setup", 1, 950.00),
    ("API Endpoint Development (20 units)", 20, 45.00),
    ("Project Management Overhead", 1, 350.00),
]

# --- 2. CALCULATIONS ---

def calculate_totals(items, tax_rate):
    """Calculates subtotal, tax amount, and grand total."""
    subtotal = sum(qty * price for _, qty, price in items)
    tax_amount = subtotal * tax_rate
    grand_total = subtotal + tax_amount
    return subtotal, tax_amount, grand_total

# calculations
SUBTOTAL, TAX_AMOUNT, GRAND_TOTAL = calculate_totals(ITEMS,
INVOICE_DETAILS["tax_rate"])

# --- 3. PDF GENERATION  ---

def create_invoice_pdf(filename="invoice.pdf"):
    """Generates the invoice PDF using ReportLab."""

    doc = SimpleDocTemplate(filename, pagesize=letter)
    styles = getSampleStyleSheet()
    story = []

    # --- Header (Company Info & Title) ---


    story.append(Paragraph(f'<font size="24"
color="#1E40AF">INVOICE</font>', styles['h1']))
    story.append(Spacer(1, 0.25*inch))


    company_text = [
        f"<b>{COMPANY_INFO['name']}</b>",
        COMPANY_INFO['address'],
        f"Phone: {COMPANY_INFO['phone']} | Email:
{COMPANY_INFO['email']}"
    ]
    for line in company_text:
        story.append(Paragraph(line, styles['Normal']))
    story.append(Spacer(1, 0.4*inch))

    # --- Invoice Details and Client Info ---

    data_table = [
        [
            Paragraph("<b>BILL TO:</b>", styles['Normal']),
            Paragraph("<b>INVOICE DETAILS:</b>", styles['Normal'])
        ],
        [
            # Column 1: Client Info

Paragraph(f"<b>{CLIENT_INFO['name']}</b><br/>{CLIENT_INFO['address']}<b
r/>Contact: {CLIENT_INFO['contact']}", styles['Normal']),
            # Column 2: Invoice Metadata
            Paragraph(f"<b>Invoice ID:</b>
{INVOICE_DETAILS['id']}<br/><b>Date:</b>
{INVOICE_DETAILS['date']}<br/><b>Due:</b>
{INVOICE_DETAILS['due_date']}", styles['Normal'])
        ]
    ]

    # Create a table for these sections (60% width each)
    t = Table(data_table, colWidths=[3*inch, 3*inch])
    t.setStyle(TableStyle([
        ('ALIGN', (0,0), (-1,-1), 'LEFT'),
        ('VALIGN', (0,0), (-1,-1), 'TOP'),
        ('BOTTOMPADDING', (0,0), (-1,0), 6),
    ]))
    story.append(t)
    story.append(Spacer(1, 0.5*inch))

    # --- Line Items Table ---

    # data for the main table (with calculated total amounts)
    item_data = [["DESCRIPTION", "QTY", "UNIT PRICE", "AMOUNT"]]

    for desc, qty, price in ITEMS:
        amount = qty * price
        item_data.append([
            desc,
            str(qty),
            f"{INVOICE_DETAILS['currency']}{price:.2f}",
            f"{INVOICE_DETAILS['currency']}{amount:.2f}"
        ])

    # Create the product table
    product_table = Table(item_data, colWidths=[3.5*inch, 0.5*inch,
1*inch, 1*inch])
    product_table.setStyle(TableStyle([
        ('BACKGROUND', (0, 0), (-1, 0), colors.HexColor('#E0E7FF')),
        ('TEXTCOLOR', (0, 0), (-1, 0), colors.black),
        ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
        ('ALIGN', (1, 1), (-1, -1), 'RIGHT'), # Align QTY, Price,
Amount to the right
        ('GRID', (0, 0), (-1, -1), 1, colors.HexColor('#D1D5DB')),
        ('FONTNAME', (0, 0), (-1, 0), 'Helvetica-Bold'),
        ('BOTTOMPADDING', (0, 0), (-1, 0), 12),
        ('BACKGROUND', (0, 1), (-1, -1), colors.white),
    ]))
    story.append(product_table)
    story.append(Spacer(1, 0.5*inch))

    # --- Totals Summary Table ---

    totals_data = [
        ["Subtotal:", f"{INVOICE_DETAILS['currency']}{SUBTOTAL:.2f}"],
        [f"Tax ({INVOICE_DETAILS['tax_rate']*100:.0f}%):",
f"{INVOICE_DETAILS['currency']}{TAX_AMOUNT:.2f}"],
        ["GRAND TOTAL:",
f"{INVOICE_DETAILS['currency']}{GRAND_TOTAL:.2f}"]
    ]

    totals_table = Table(totals_data, colWidths=[1.5*inch, 1*inch])
    totals_table.setStyle(TableStyle([
        ('ALIGN', (0, 0), (-1, -1), 'RIGHT'),
        ('FONTNAME', (0, 2), (-1, 2), 'Helvetica-Bold'), # Bold the
Total line
        ('FONTSIZE', (0, 2), (-1, 2), 14), # Larger Total font
        ('TOPPADDING', (0, 2), (-1, 2), 6),
        ('LINEBELOW', (0, 1), (-1, 1), 1, colors.black), # Line above
Total
    ]))

    #  totals table in a full-width table to push it to the right
    wrapper_table = Table([
        ["", totals_table]
    ], colWidths=[4*inch, 2.5*inch])
    wrapper_table.setStyle(TableStyle([
        ('ALIGN', (1, 0), (1, 0), 'RIGHT'), # Align the totals table to
the right
        ('LEFTPADDING', (0, 0), (0, 0), 0),
        ('RIGHTPADDING', (-1, -1), (-1, -1), 0),
    ]))
    story.append(wrapper_table)

    # --- Footer ---
    story.append(Spacer(1, 0.75*inch))
    footer_text = "Thank you for your business. Please make payment by
the due date."
    story.append(Paragraph(f'<div align="center">{footer_text}</div>',
styles['Normal']))

    # Build the final PDF document
    doc.build(story)
    print(f"\n--- SUCCESS ---")
    print(f"Invoice generated successfully as '{filename}'")
    print(f"Total Amount Due:
{INVOICE_DETAILS['currency']}{GRAND_TOTAL:.2f}")

# --- EXECUTION ---
if __name__ == '__main__':
    create_invoice_pdf()
```

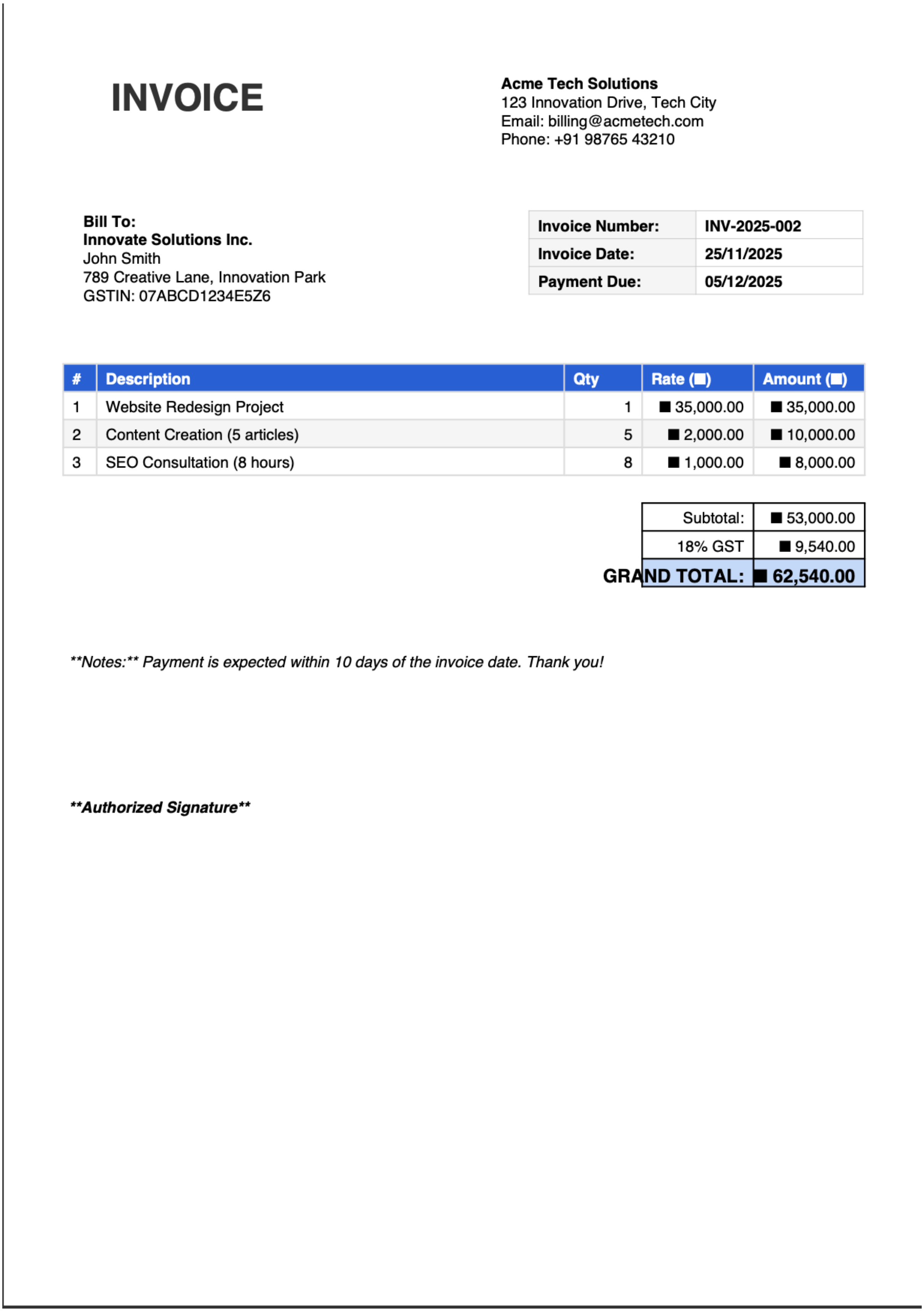
OUTPUT:



```
        ('TOPPADDING', (0, 2), (-1, 2), 6),
        ('LINEBELOW', (0, 1), (-1, 1), 1, colors.black), # Line above Total
    ]))

    #  totals table in a full-width table to push it to the right
    wrapper_table = Table([
        ["", totals_table]
    ], colWidths=[4*inch, 2.5*inch])
    wrapper_table.setStyle(TableStyle([
        ('ALIGN', (1, 0), (1, 0), 'RIGHT'), # Align the totals table to the right
        ('LEFTPADDING', (0, 0), (0, 0), 0),
        ('RIGHTPADDING', (-1, -1), (-1, -1), 0),
    ]))
    story.append(wrapper_table)

    # --- Footer ---
    story.append(Spacer(1, 0.75*inch))
    footer_text = "Thank you for your business. Please make payment by the due date."
    story.append(Paragraph(f'<div align="center">{footer_text}</div>', styles['Normal']))

    # Build the final PDF document
    doc.build(story)
    print(f"\n--- SUCCESS ---")
    print(f"Invoice generated successfully as '{filename}'")
    print(f"Total Amount Due: {INVOICE_DETAILS['currency']}{GRAND_TOTAL:.2f}")

# --- EXECUTION ---
if __name__ == '__main__':
    create_invoice_pdf()
```

```
--- SUCCESS ---
Invoice generated successfully as 'invoice.pdf'
Total Amount Due: $5720.00
```

PDF:



INVOICE

Acme Tech Solutions
123 Innovation Drive, Tech City
Email: billing@acmetech.com
Phone: +91 98765 43210

Bill To:
Innovate Solutions Inc.
John Smith
789 Creative Lane, Innovation Park
GSTIN: 07ABCD1234E5Z6

Invoice Number:	**INV-2025-002**
Invoice Date:	**25/11/2025**
Payment Due:	**05/12/2025**

#	Description	Qty	Rate (■)	Amount (■)
1	Website Redesign Project	1	■ 35,000.00	■ 35,000.00
2	Content Creation (5 articles)	5	■ 2,000.00	■ 10,000.00
3	SEO Consultation (8 hours)	8	■ 1,000.00	■ 8,000.00

Subtotal:	■ 53,000.00
18% GST	■ 9,540.00
GRAND TOTAL:	**■ 62,540.00**

***Notes:** Payment is expected within 10 days of the invoice date. Thank you!*

*****Authorized Signature*****

PDF Link:

https://drive.google.com/file/d/1-mhSk2i1Mptq4Kmj9Ujwo3FOxBokxjvX/view?usp=drive_link