# INVOICE GENERATOR

**A MINI PROJECT REPORT**

**18CSC207J - ADVANCED PROGRAMMING PRACTICE**

*Submitted by*

**KOTIKALAPUDI TULASI VENKATA BHADRA GANESH (RA2111003011769)**
**S VAMSI VARSHITH(RA2111003011782)**

*Under the guidance of*

**DR.G.BALAMURUGAN**

Assistant Professor, Department of Computer Science and Engineering

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**


S.R.M. Nagar, Kattankulathur, Chengalpattu District

**MAY 2023**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **"INVOICE GENERATOR"** is the bonafide Work of KOTIKALAPUDI TULASI VENKATA BHADRA GANESH (RA2111003011769),S VAMSI VARSHITH(RA2111003011782) who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

DR.G.BALAMURUGAN
**FACULTY INCHARGE**
Assistant Professor
Department of Computing Technologies

**SIGNATURE**

Dr. M.PUSHPALATHA
**HEAD OF THE DEPARTMENT**
Professor & Head
Department of Computing Technologies

# ABSTRACT

An invoice generator is a software application that automates the process of creating invoices for businesses. The generator takes inputs such as customer information, product or service details, pricing, and taxes, and generates an invoice in a standardized format that can be sent to customers. The advantages of using an invoice generator include increased efficiency, accuracy, and consistency in generating invoices. In addition, it can help businesses manage their cash flow and provide better customer service by providing timely and accurate invoices. This abstract provides a brief overview of an invoice generator and highlights some of its benefits

# TABLE OF CONTENTS

# CHAPTER 1

## INTRODUCTION

An invoice is a vital part of any business transaction as it serves as a formal document that outlines the details of a sale, including the products or services provided, the cost of those items, and any applicable taxes or discounts. Generating an invoice can be a time-consuming and error-prone process, especially for businesses with a high volume of transactions. To address this issue, many businesses are turning to invoice generator software to automate the process and improve efficiency.

An invoice generator is a software tool that streamlines the invoicing process by automating the creation of invoices. It takes the information about the sale, including the customer's information, product or service details, pricing, and taxes, and generates a standardized invoice that can be sent to the customer. This not only saves time but also reduces the risk of errors and inconsistencies.

In this article, we will explore the benefits of using an invoice generator, how it works, and some popular invoice generator software options available in the market.

# CHAPTER 2

# METHODOLOGY

The methodology for an invoice generator involves several steps to ensure that the software tool functions efficiently and produces accurate invoices. The following are some of the key steps involved in the methodology for an invoice generator:

Gathering requirements: The first step in developing an invoice generator is to gather requirements from stakeholders, including the finance and sales teams. This involves understanding their needs, such as the information that needs to be included on invoices, the type of invoice templates required, and the preferred format for invoices.

Designing the software: Based on the requirements, the software is designed to automate the process of generating invoices. The design includes defining the data models, the invoice templates, and the algorithms for generating invoices.

Developing the software: Once the design is complete, the software development process begins. This involves coding the software and building the user interface for the invoice generator.

Testing and debugging: The software is tested extensively to ensure that it functions correctly and produces accurate invoices. Any bugs or errors that are identified during testing are fixed.

Integration and deployment: Once the software has been tested and debugged, it is integrated with other systems and deployed in the production environment.

Ongoing maintenance and support: The software is monitored and maintained to ensure that it continues to function correctly. Support is provided to users to address any issues they may encounter while using the software.

By following this methodology, an invoice generator can be developed and deployed effectively to automate the process of generating invoices for a business.

# CHAPTER 3

# CODING AND TESTING

## CODE:

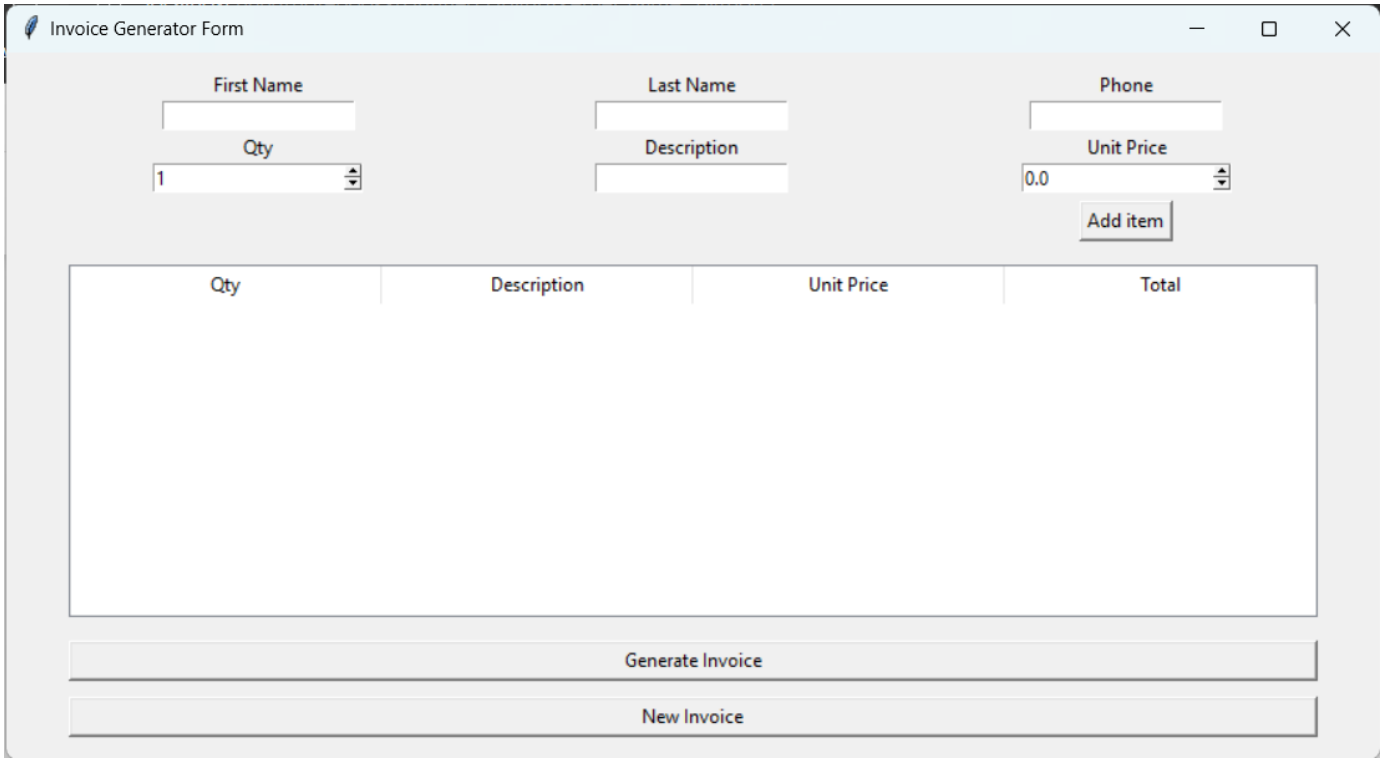
```
!pip install docxtpl
import tkinter
from tkinter import ttk
from docxtpl import DocxTemplate
import datetime
from tkinter import messagebox

def clear_item():
    qty_spinbox.delete(0, tkinter.END)
    qty_spinbox.insert(0, "1")
    desc_entry.delete(0, tkinter.END)
    price_spinbox.delete(0, tkinter.END)
    price_spinbox.insert(0, "0.0")

invoice_list = []
def add_item():
    qty = int(qty_spinbox.get())
    desc = desc_entry.get()
    price = float(price_spinbox.get())
    line_total = qty*price
    invoice_item = [qty, desc, price, line_total]
    tree.insert('',0, values=invoice_item)
    clear_item()

    invoice_list.append(invoice_item)


def new_invoice():
    first_name_entry.delete(0, tkinter.END)
    last_name_entry.delete(0, tkinter.END)
    phone_entry.delete(0, tkinter.END)
    clear_item()
    tree.delete(*tree.get_children())

    invoice_list.clear()
```

```
def generate_invoice():
    doc = DocxTemplate("invoice_template.docx")
    name = first_name_entry.get()+last_name_entry.get()
    phone = phone_entry.get()
    subtotal = sum(item[3] for item in invoice_list)
    salestax = 0.1
    total = subtotal*(1-salestax)

    doc.render({"name":name,
            "phone":phone,
            "invoice_list": invoice_list,
            "subtotal":subtotal,
            "salestax":str(salestax*100)+"%",
            "total":total})

    doc_name = "new_invoice" + name + datetime.datetime.now().strftime("%Y-%m-%d-%H%M%S") +
".docx"
    doc.save(doc_name)

    messagebox.showinfo("Invoice Complete", "Invoice Complete")

    new_invoice()



window = tkinter.Tk()
window.title("Invoice Generator Form")

frame = tkinter.Frame(window)
frame.pack(padx=20, pady=10)

first_name_label = tkinter.Label(frame, text="First Name")
first_name_label.grid(row=0, column=0)
last_name_label = tkinter.Label(frame, text="Last Name")
last_name_label.grid(row=0, column=1)

first_name_entry = tkinter.Entry(frame)
last_name_entry = tkinter.Entry(frame)
first_name_entry.grid(row=1, column=0)
last_name_entry.grid(row=1, column=1)
```

```
phone_label = tkinter.Label(frame, text="Phone")
phone_label.grid(row=0, column=2)
phone_entry = tkinter.Entry(frame)
phone_entry.grid(row=1, column=2)

qty_label = tkinter.Label(frame, text="Qty")
qty_label.grid(row=2, column=0)
qty_spinbox = tkinter.Spinbox(frame, from_=1, to=100)
qty_spinbox.grid(row=3, column=0)

desc_label = tkinter.Label(frame, text="Description")
desc_label.grid(row=2, column=1)
desc_entry = tkinter.Entry(frame)
desc_entry.grid(row=3, column=1)

price_label = tkinter.Label(frame, text="Unit Price")
price_label.grid(row=2, column=2)
price_spinbox = tkinter.Spinbox(frame, from_=0.0, to=500, increment=0.5)
price_spinbox.grid(row=3, column=2)

add_item_button = tkinter.Button(frame, text = "Add item", command = add_item)
add_item_button.grid(row=4, column=2, pady=5)

columns = ('qty', 'desc', 'price', 'total')
tree = ttk.Treeview(frame, columns=columns, show="headings")
tree.heading('qty', text='Qty')
tree.heading('desc', text='Description')
tree.heading('price', text='Unit Price')
tree.heading('total', text="Total")


tree.grid(row=5, column=0, columnspan=3, padx=20, pady=10)


save_invoice_button = tkinter.Button(frame, text="Generate Invoice", command=generate_invoice)
save_invoice_button.grid(row=6, column=0, columnspan=3, sticky="news", padx=20, pady=5)
new_invoice_button = tkinter.Button(frame, text="New Invoice", command=new_invoice)
new_invoice_button.grid(row=7, column=0, columnspan=3, sticky="news", padx=20, pady=5)


window.mainloop()
```

# CHAPTER 4

# SCREENSHOTS AND RESULTS



# CHAPTER 5

## CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, an invoice generator is a valuable tool for businesses that need to streamline their invoicing process. It can save time and reduce the risk of errors, ensuring that invoices are accurate and sent out in a timely manner. The methodology for developing an invoice generator involves several steps, including gathering requirements, designing the software, developing, testing, and deploying it.

There are many potential future enhancements for an invoice generator. Some of these may include:

Integration with accounting software: An invoice generator can be integrated with accounting software to automate the bookkeeping process.

Automated payment reminders: The invoice generator could include a feature to automatically send payment reminders to customers who have outstanding balances.

Multi-currency support: The software could support multiple currencies, making it easier for businesses to invoice customers in different countries.

Mobile app support: A mobile app could be developed for the invoice generator, allowing users to generate invoices on the go.

Machine learning-based invoicing: Machine learning algorithms could be integrated with the software to automatically suggest pricing, discounts, and other details based on past transactions.

Overall, an invoice generator can be a valuable tool for businesses of all sizes. By automating the process of generating invoices, it can save time and reduce the risk of errors, helping businesses to improve their cash flow and customer service.

# REFERENCES

References:

Zoho Invoice: https://www.zoho.com/invoice/

QuickBooks Online: https://quickbooks.intuit.com/online/

FreshBooks: https://www.freshbooks.com/

Wave: https://www.waveapps.com/invoicing/

Xero: https://www.xero.com/us/invoicing/

PayPal: https://www.paypal.com/invoicing

Invoice Ninja: https://www.invoiceninja.com/

Invoicely: https://invoicely.com/

Square Invoices: https://squareup.com/invoices

Hiveage: https://www.hiveage.com/