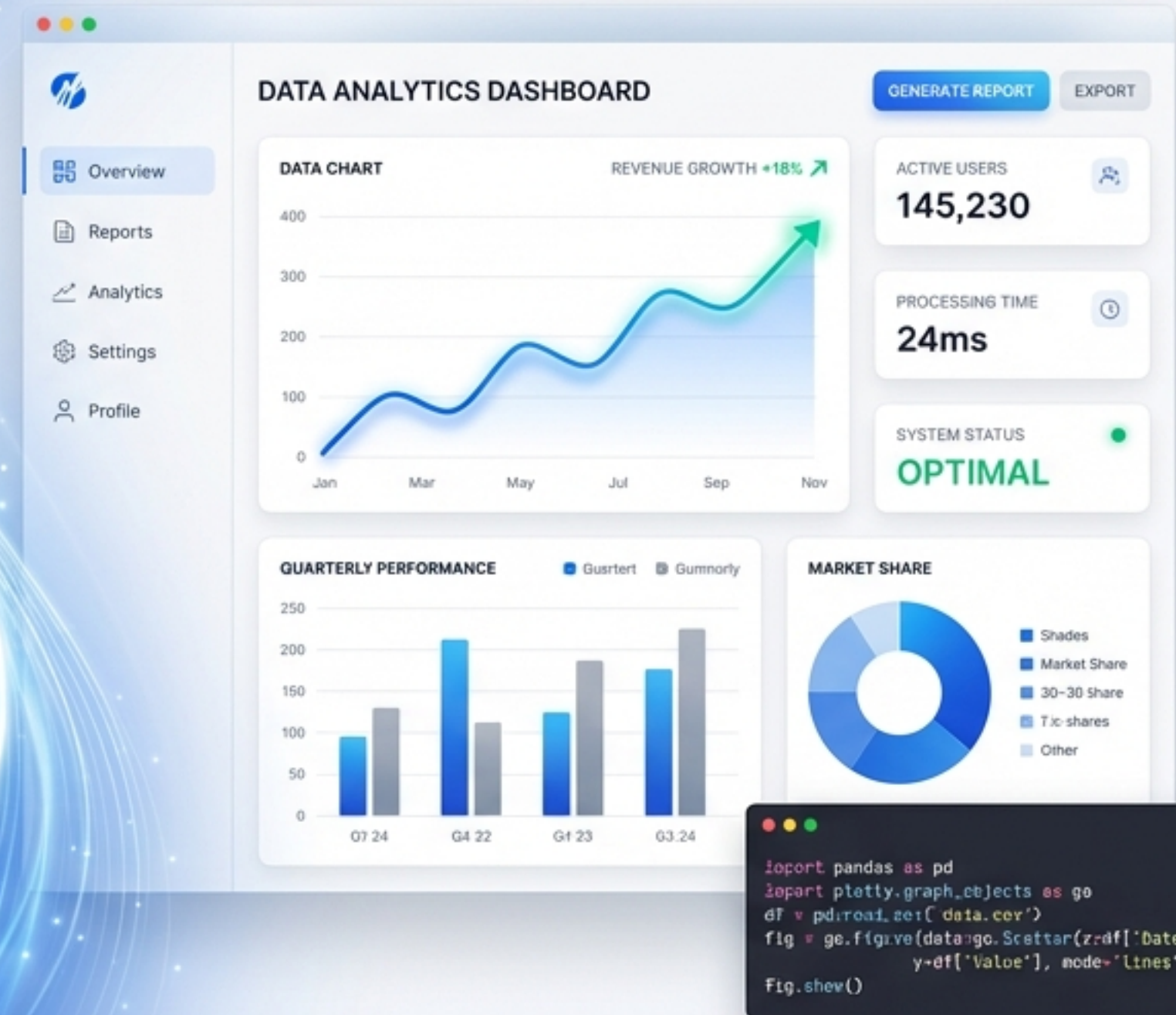


"From Blueprint to Masterpiece: Crafting Modern, Data-Driven GUIs with Python"



The Foundation: Capturing Structured Data

Student Data Entry Form

Name:

Class:

Three

▼

Mark:

Gender:

☐ Male

☒ Female

☐ Others

Add Record

```
# A simple Combobox for class selection
options = ["Three", "Four", "Five"]
cb_class = ttk.Combobox(wy_w, values=options)

# Radiobuttons for gender
radio_v = tk.StringVar(value="Female")
r1 = tk.Radiobutton(my_w, text="Male", variable=radio_v, value="Male")
r2 = tk.Radiobutton(wy_w, text="Female", variable=radio_v, value="Female")
```

Intelligent Input: The Autocomplete Advantage

Standard Input

Name:

Autocomplete Input

Name:

John Deo ✓

Connect to the database
`from sqlalchemy import create_engine`
`my_conn = create_engine("mysql+mysqldb://id:pw@localhost/my_tutorial")`

Query for autocomplete options
`q = "SELECT name FROM student"`
`my_cursor = my_conn.execute(q)`
`result = my_cursor.fetchall()`
`my_list = [r for r, in result]`

Assembling the Core: The Invoice Display

Product:  Quantity:  

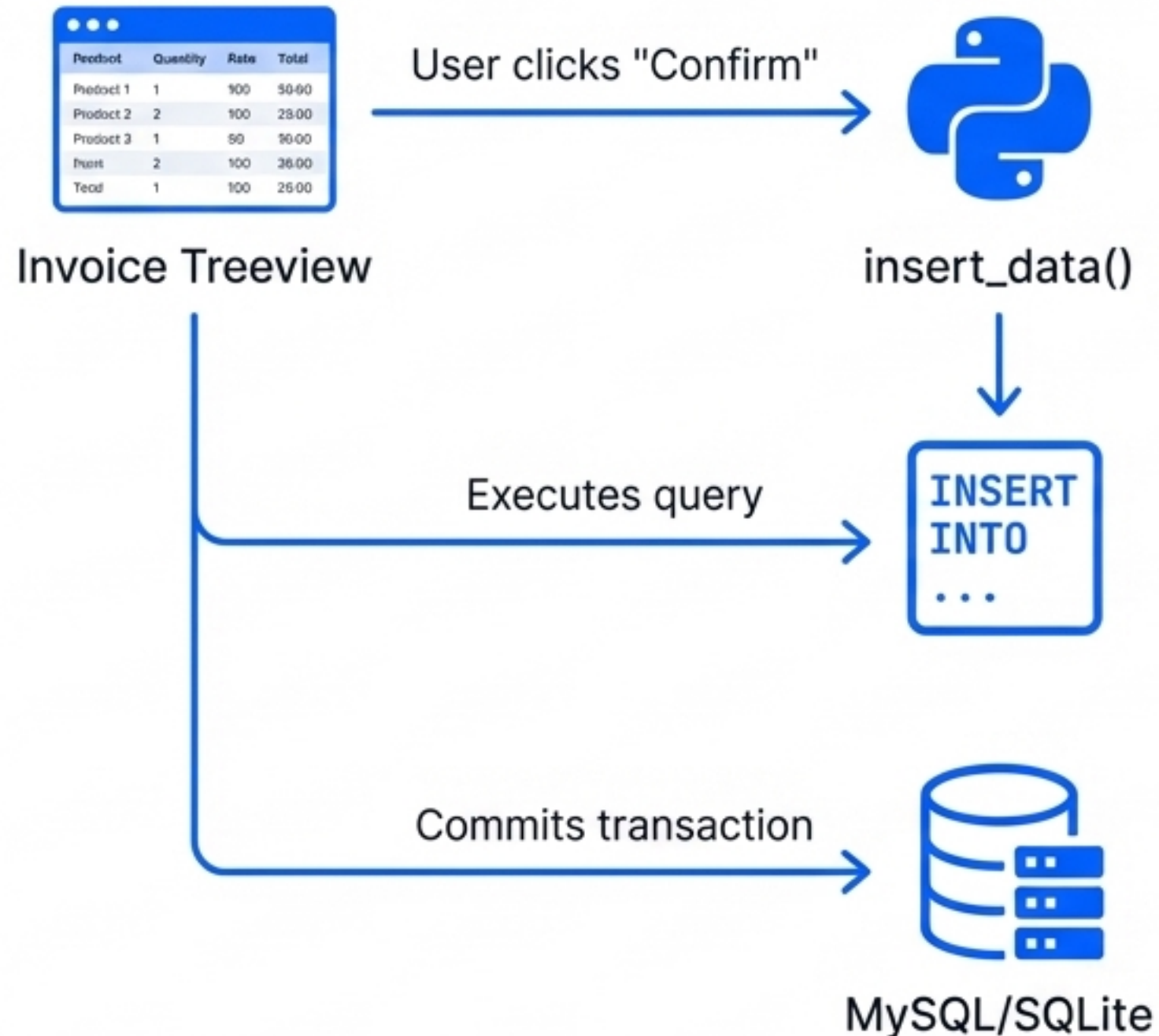
Sl No	Product	Quantity	Rate	Total
1	Monitor	2	150.00	300.00
2	Keyboard	2	25.00	50.00
3	Mouse	3	10.00	30.00

Sub-Total: 380.00
Tax: 38.00
Grand Total: 418.00

```
# Add a new item to the Treeview
iid = 4
product_name = "Pen Drive"
qty = 5
rate = 12.00
total = qty * rate

trv.insert("", 'end', iid=iid, values=(
    iid,
    product_name,
    qty,
    rate,
    total
))
```

Persistence: Saving Invoices to the Database



Success!

Bill No: 101, Products: 4

```
def insert_data():  
    # ...  
    query = "INSERT INTO plus2_invoice (total, dt) VALUES  
    id = my_conn.execute(query, data)  
    inv_id = id.lastrowid  
    # ...  
    query_dtl = "INSERT INTO plus2_invoice_dtl (inv_id, .  
    my_conn.execute(query_dtl, my_data)
```

The First Polish: Modernizing User Experience

BEFORE

VS.

AFTER



<

October 2023

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

```
de = ttk.DateEntry(dateformat='%Y-%m-%d', bootstyle='primary')
```

primary

2023-10-27

success

2023-10-27

info

2023-10-27

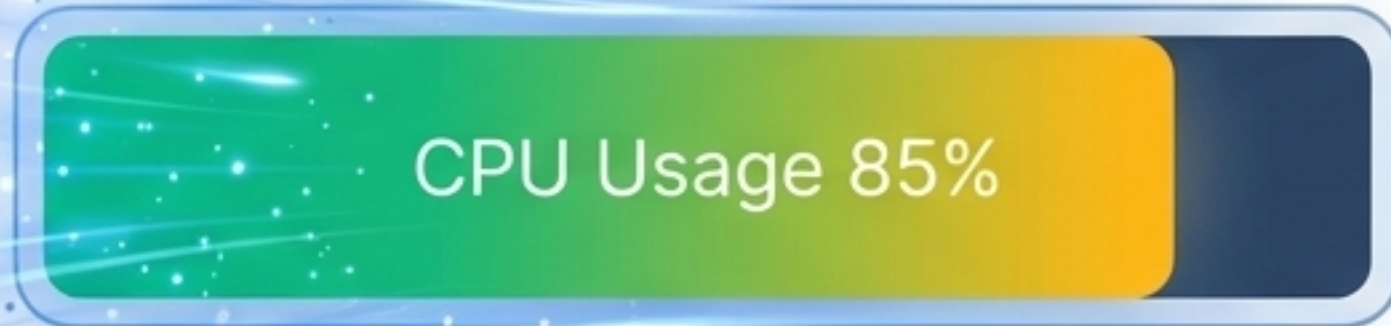
warning

2023-10-27

danger

2023-10-27

Real-Time Feedback: Visualizing Progress



```
# Create a shared variable
progress_var = ttk.IntVar()

# Link the Floodgauge to the variable
fg = ttk.Floodgauge(
    variable=progress_var,
    bootstyle=SUCCESS,
    mask='CPU Usage {}%'
)

# Link another widget, like a Scale,
# to the same variable for synchronization
scale = ttk.Scale(variable=progress_var)
```

The Transformation: From Functional to Phenomenal

BEFORE

Invoice Application

File Menu Help

Customer Name

Invoice Number

Date

Item Description

Add Item

Calculate

Quantity

Price

Total

Save Invoice

plus2net

Before
`my_w = tk.Tk()`



After

`my_w = ttk.Window(themename="superhero")`

AFTER

Invoice Application

File Menu Help

Customer Name

Date

Invoice Number

Avoice Date

Item Description

Add Item

Calculate

Quantity

Price

Total

Save Invoice

Final Touches: Adding Modern Utility

superhero

superhero

Customer Name	Invoice Number	Date

Item Description	Quantity
	Price

Add Item Calculate

Total

Generate QR Code

QR Code



```
import pyqrcode
from PIL import Image

# Create the QR code from input
my_qr = pyqrcode.create(e1.get())

# ... code to embed logo ...

# Display in a Tkinter label
my_qr = my_qr.xbm(scale=10)
my_img = tk.BitmapImage(data=my_qr)
l1.config(image=my_img)
```

Your Blueprint for Success



Structure First: Build a solid application core with standard Tkinter and ttk widgets like Entry, Treeview, and Spinbox.



Connect Your Data: Power your UI with live data using SQLAlchemy for robust, real-time database integration.



Elevate with ttkbootstrap: Instantly transform your application's appearance and user experience with modern themes and advanced widgets like DateEntry and Floodgauge.

www.plus2net.com



<https://www.youtube.com/@plus2net1>

<https://www.linkedin.com/in/plus2net/>

<https://x.com/plus2net>