```python
import tkinter as tk
from tkinter import messagebox
import hashlib
import time

# ---------------- Dummy Blockchain ---------------- #
class Block:
    def __init__(self, index, value, prev_hash):
        self.index = index
        self.value = value
        self.timestamp = time.strftime("%Y-%m-%d %H:%M:%S")
        self.prev_hash = prev_hash
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        block_data = f"{self.index}{self.value}{self.timestamp}{self.prev_hash}"
        return hashlib.sha256(block_data.encode()).hexdigest()

class DummyBlockchain:
    def __init__(self):
        self.chain = []
        self.create_genesis_block()

    def create_genesis_block(self):
        genesis = Block(0, 0, "0")
        self.chain.append(genesis)

    def add_block(self, value):
        prev_block = self.chain[-1]
        new_block = Block(len(self.chain), value, prev_block.hash)
        self.chain.append(new_block)

    def get_latest_value(self):
        return self.chain[-1].value

# ---------------- Smart Contract Simulation ---------------- #
class SimpleStorageContract:
    def __init__(self, blockchain):
        self.blockchain = blockchain

    def set(self, value):
        self.blockchain.add_block(value)

    def get(self):
        return self.blockchain.get_latest_value()

# ---------------- Tkinter GUI ---------------- #
blockchain = DummyBlockchain()
contract = SimpleStorageContract(blockchain)

def set_value():
    try:
        value = int(entry.get())
        contract.set(value)
        messagebox.showinfo("Transaction Successful", "Value stored on dummy blockchain!")
        entry.delete(0, tk.END)
        update_chain_display()
    except ValueError:
        messagebox.showerror("Error", "Please enter a valid integer")

def get_value():
    value = contract.get()
    messagebox.showinfo("Stored Value", f"Current Stored Value: {value}")

def update_chain_display():
    chain_text.delete("1.0", tk.END)
    for block in blockchain.chain:
        chain_text.insert(tk.END,
            f"Block #{block.index}\n"
            f"Stored Value: {block.value}\n"
            f"Timestamp: {block.timestamp}\n"
            f"Prev Hash: {block.prev_hash}\n"
            f"Hash: {block.hash}\n"
            f"{'-'*40}\n"
        )

# GUI Window
root = tk.Tk()
root.title("Simple Storage Smart Contract (Dummy Blockchain)")
root.geometry("650x600")

tk.Label(root, text="Simple Storage Smart Contract", font=("Arial", 16, "bold")).pack(pady=10)

frame = tk.Frame(root)
frame.pack(pady=10)

tk.Label(frame, text="Enter Value: ").grid(row=0, column=0)
entry = tk.Entry(frame)
entry.grid(row=0, column=1)

tk.Button(frame, text="Set Value (Transaction)", command=set_value).grid(row=1, column=0,
pady=5)
tk.Button(frame, text="Get Value", command=get_value).grid(row=1, column=1, pady=5)

tk.Label(root, text="Dummy Blockchain Ledger", font=("Arial", 12, "bold")).pack(pady=10)

chain_text = tk.Text(root, height=20, width=75)
chain_text.pack()

update_chain_display()
root.mainloop()

```