from Npp import \*

import re

# --- Sample Lookup Table ---

lookup\_table = [

{"Code": "12", "State": "NY", "Session": "East"},

{"Code": "34", "State": "TX", "Session": "Texas"},

{"Code": "56", "State": "CA", "Session": "West"},

# Add your actual codes/states/sessions here

]

line\_business\_options = [

"Artisan Contractor", "Auto Service & Repair", "Commercial Auto",

"Habitational", "Manufacturing", "Real Estate", "Restaurant",

"Retail", "Umbrella", "Wholesale Distribution"

]

def get\_session\_from\_input(value):

value = value.strip()

for row in lookup\_table:

if value.isdigit() and row['Code'] == value:

return row['Session']

elif value.isalpha() and row['State'].upper() == value.upper():

return row['Session']

return None

def insert\_or\_replace\_line(prefix, value, after\_line):

for i in range(editor.getLineCount()):

if editor.getLine(i).startswith(prefix):

editor.replaceWholeLine(i, f"{prefix} {value}\n")

return

editor.insertText(editor.positionFromLine(after\_line + 1), f"{prefix} {value}\n")

def line\_index(label):

lines = [editor.getLine(i).strip() for i in range(editor.getLineCount())]

return next((i for i, l in enumerate(lines) if l.startswith(label)), -1)

def on\_charadded(char):

lines = [editor.getLine(i).strip() for i in range(editor.getLineCount())]

policy\_i = line\_index("Policy:")

policy\_val = lines[policy\_i][len("Policy:"):].strip() if policy\_i != -1 else ""

if policy\_val:

if line\_index("Policy Status:") == -1:

insert\_or\_replace\_line("Policy Status:", "", policy\_i)

if line\_index("Line of Business:") == -1:

lob\_line = "Line of Business: [Select from: " + ", ".join(line\_business\_options) + "]"

editor.insertText(editor.positionFromLine(policy\_i + 2), lob\_line + "\n")

if line\_index("Session:") == -1:

insert\_or\_replace\_line("Session:", "", policy\_i + 3)

# Validate Session and resolve it

session\_i = line\_index("Session:")

if session\_i != -1:

session\_line = lines[session\_i]

session\_input = session\_line[len("Session:"):].strip()

if session\_input and not session\_line.endswith("]"):

resolved = get\_session\_from\_input(session\_input)

if resolved:

editor.replaceWholeLine(session\_i, f"Session: {session\_input} [{resolved}]\n")

# Show Q1–Q5 once all required fields are filled

required\_labels = ["SR#:", "Policy:", "Policy Status:", "Line of Business:", "Session:"]

all\_ready = all(line\_index(lbl) != -1 and lines[line\_index(lbl)].split(":", 1)[-1].strip() for lbl in required\_labels)

if all\_ready and line\_index("1. Who is requesting?") == -1:

editor.insertText(editor.getLength(), "\n1. Who is requesting? \n- Type: [Agent / Agency Rep / Insured]\n\n2. What is being requested?\n\n3. Is this an Endorsement or a BO/RI?\n\n4. What is the Effective Date?\n5. Does it make sense?\n")

# Handle Licensed field

type\_i = line\_index("- Type:")

if type\_i != -1:

type\_val = lines[type\_i].split(":", 1)[-1].strip()

if type\_val in ["Agent", "Agency Rep"] and line\_index("- Licensed:") == -1:

editor.insertText(editor.positionFromLine(type\_i + 1), "- Licensed: [Yes / No]\n")

# Update Q1 with suffix

q1\_i = line\_index("1. Who is requesting?")

if q1\_i != -1:

q1\_line = lines[q1\_i]

name\_match = re.match(r"1\. Who is requesting\?\s+(.+?)(?:\s+-\s+.\*)?$", q1\_line)

if name\_match:

name = name\_match.group(1)

type\_val = lines[type\_i].split(":", 1)[-1].strip() if type\_i != -1 else ""

licensed\_val = lines[line\_index("- Licensed:")].split(":", 1)[-1].strip() if line\_index("- Licensed:") != -1 else ""

if type\_val == "Agent":

suffix = " - Licensed Agent" if licensed\_val == "Yes" else " - Agent"

elif type\_val == "Agency Rep":

suffix = " - Licensed Agency Rep" if licensed\_val == "Yes" else " - Agency Rep"

elif type\_val == "Insured":

suffix = " - Insured"

else:

suffix = ""

editor.replaceWholeLine(q1\_i, f"1. Who is requesting? {name}{suffix}\n")

def on\_bufferactivated(args):

editor.callback(on\_charadded, [SCINTILLANOTIFICATION.CHARADDED])

# 🔹 Insert Initial Form

editor.setText("SR#:\n\nPolicy:\n")

# 🔹 Activate listener

notepad.callback(on\_bufferactivated, [NOTIFICATION.BUFFERACTIVATED])

editor.callback(on\_charadded, [SCINTILLANOTIFICATION.CHARADDED])