

Covid-19

**20213939 Ahmed Yasser Ahmed**

**Supervised by: PROF. Osama Fathy**

**TA: ENG. Ola Mahmoud**

**C5.2**

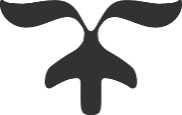


Table of Contents

[**I.** **Introduction** 1](#_Toc197299763)

[**II.** **Analysis of the Project** 2](#_Toc197299764)

[**III.** **Identification Problem** 3](#_Toc197299765)

[**IV.** **Decision tree** 5](#_Toc197299766)

[**V.** **Production Rules** 6](#_Toc197299767)

[**VI.** **Object-Attribute-Value (OAV)** 7](#_Toc197299768)

[**VII.** **Semantic Network** 10](#_Toc197299769)

[**VIII.** **Frames** 11](#_Toc197299770)

[**IX.** **Project Implementation [Code]** 12](#_Toc197299771)

[**X.** **Conclusion** 25](#_Toc197299772)

[References 25](#_Toc197299773)

# **Introduction**

The COVID-19 pandemic has posed a global health crisis, making early detection and risk evaluation crucial. Our project introduces a COVID-19 Expert System developed using Python and the experta rule-based library. It simulates a diagnostic consultation by asking a series of structured questions based on the user's symptoms, age, and medical history. The goal is to provide a basic risk level assessment and advise the user accordingly. This type of system demonstrates how expert systems can assist in medical triage and reduce the burden on healthcare systems.

# **Analysis of the Project**

The project was implemented using the experta library, a Python framework for building expert systems using the CLIPS rule-based paradigm. The system includes eight diagnostic levels, each with 3 questions, totaling 24 inputs.

**Each level gathers specific information:**

* **Levels 1–4:** Focus on symptoms such as fever, cough, and loss of taste/smell.
* **Levels 5–6:** Target risk factors including age and pre-existing conditions.
* **Levels 7–8:** Analyze the duration and severity of symptoms and exposure history.

**Knowledge Representation Techniques**

* **Decision Tree:** A hierarchical structure was used where each level acts like a node that decides the path based on user inputs.
* **Production Rules:** Implemented via @Rule decorators in the experta framework. Each rule maps to a combination of user inputs to determine risk.
* **Object-Attribute-Value (OAV)**: The Fact object stores attributes like symptom1='yes', maintaining clear state representation.
* **Semantic Network:** Concepts like "symptoms", "risk factors", and "exposure" are interconnected via rule relationships.
* **Frames:** Each level can be viewed as a frame with slots (question responses) representing the user's state.
* **System Workflow:**
  1. Accepts user input for each question.
  2. Stores responses in Fact objects.
  3. Matches rules in the knowledge base.
  4. Infers the risk level (Low, Medium, High).
  5. Displays a recommendation based on inference.

# **Identification Problem**

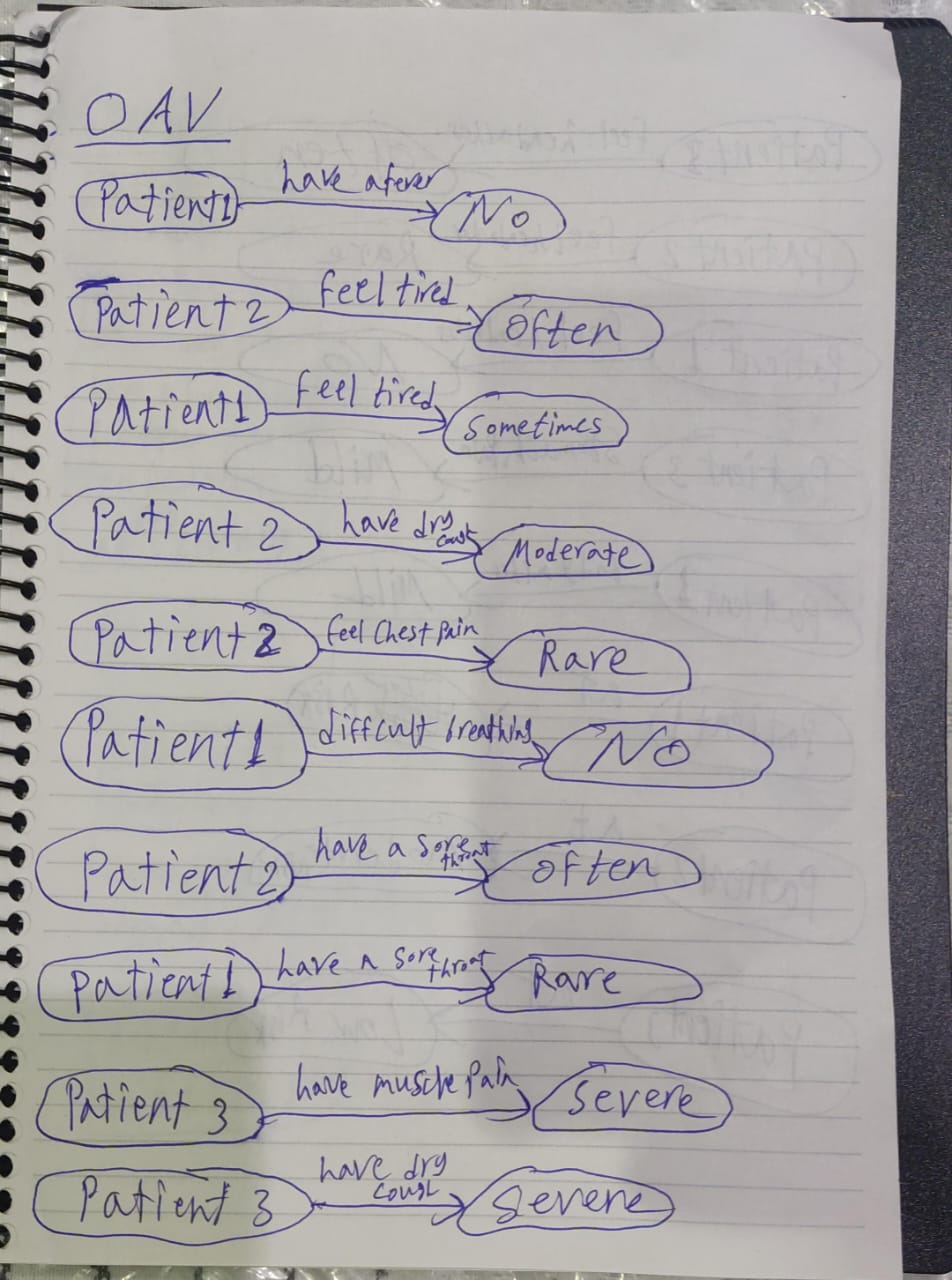
|  |  |  |  |
| --- | --- | --- | --- |
| Attribute/Type | Low Risk | Moderate Risk | High Risk |
| General Symptoms | **Moderate** | **Moderate** | **High** |
| Breathing and Chest Symptoms | **Sometimes** | **Often** | **Always** |
| Sensory Symptoms | **No** | **Rare** | **Severe** |
| Pain and Aches Symptoms | **Slight** | **Moderate** | **Always** |
| Digestive Symptoms | **Mild** | **Moderate** | **Severe** |
| Mental Symptoms | **No** | **Sometimes** | **Often** |
| Pre-existing Conditions | **Controlled** | **Uncontrolled** | **Severe** |
| Risk Factors | **No** | **Moderate** | **High** |

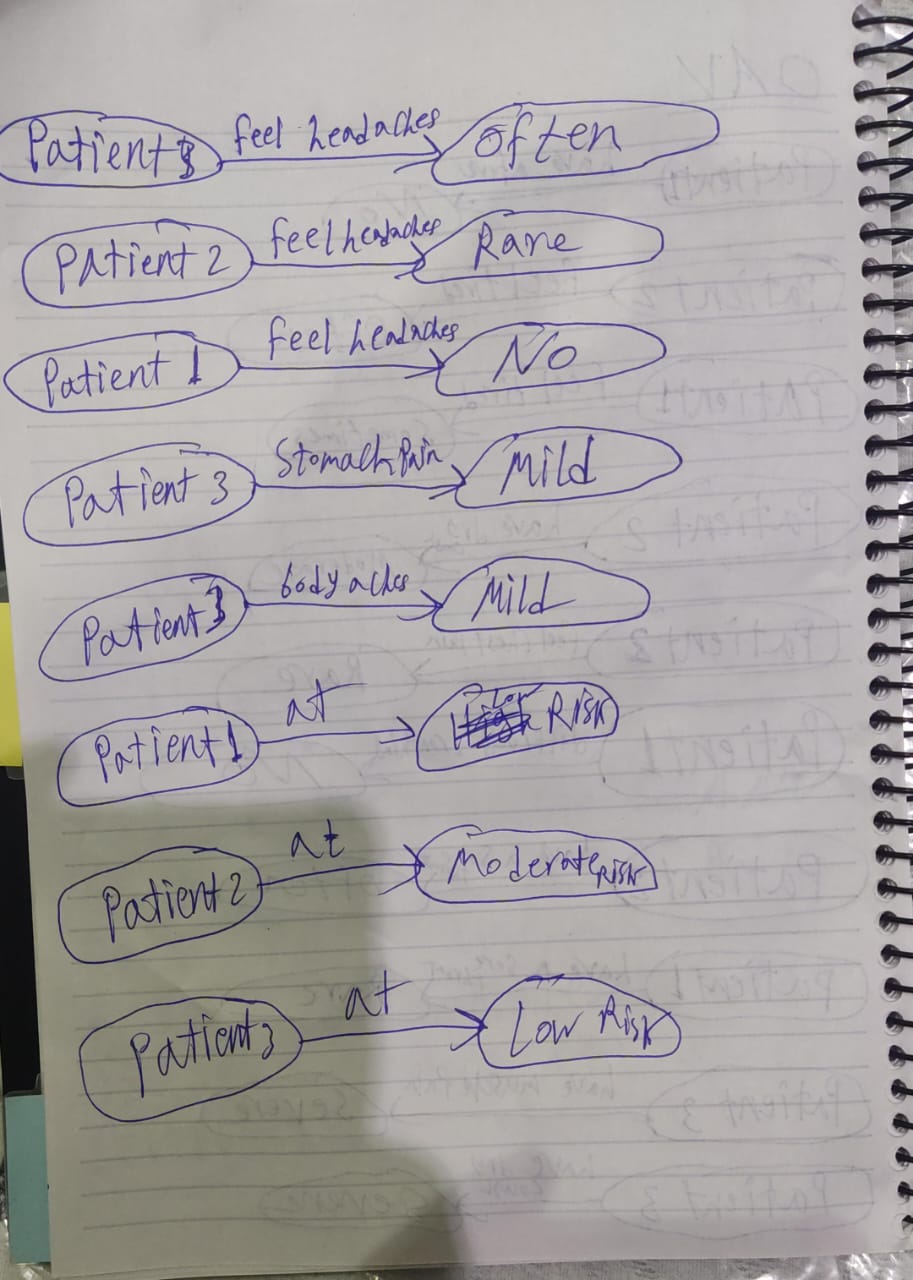
# A paper with writing on it**Decision tree**

# A notebook with writing on it AI-generated content may be incorrect.**Production Rules**

# **Object-Attribute-Value (OAV)**

|  |  |  |
| --- | --- | --- |
| Object | Attribute | Value |
| Patient 1 | Have a fever | No |
| Patient 2 | Feel tired | Often |
| Patient 1 | Feel tired | Sometimes |
| Patient 2 | Have dry cough | Moderate |
| Patient 2 | feel chest pain | Rare |
| Patient 1 | Difficulty breathing | No |
| Patient 2 | Have a Sore Throat | Often |
| Patient 1 | Have a Sore Throat | Rare |
| Patient 3 | Have a muscle pain | Severe |
| Patient 3 | Have dry cough | Severe |
| Patient 3 | Feel headaches | Often |
| Patient 2 | Feel headaches | Rare |
| Patient 1 | Feel head aches | No |
| Patient 3 | Stomach | Mild |
| Patient 3 | Body aches | Mild |
| Patient 1 | At | Low Risk |
| Patient 2 | At | Moderate Risk |
| Patient 3 | At | High Risk |





# A paper with writing on it AI-generated content may be incorrect.**Semantic Network**

# A notebook with writing on it AI-generated content may be incorrect.**Frames**

# **Project Implementation [Code]**

The implementation used Python 3 and the experta library. Key components include:

**Libraries:**

* experta: For defining facts and rules.
* os: To clear the console at each stage.
* time: To simulate processing delay.

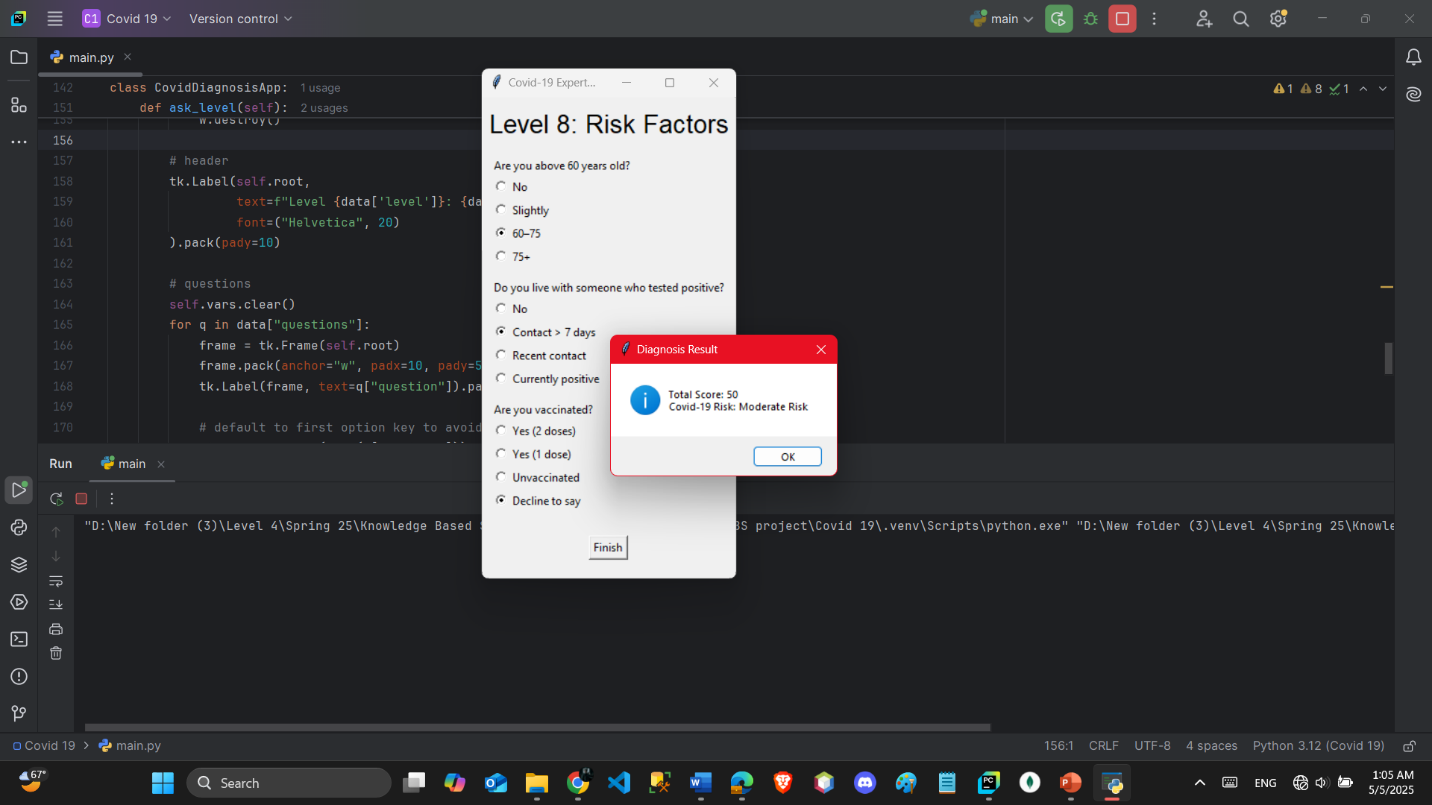
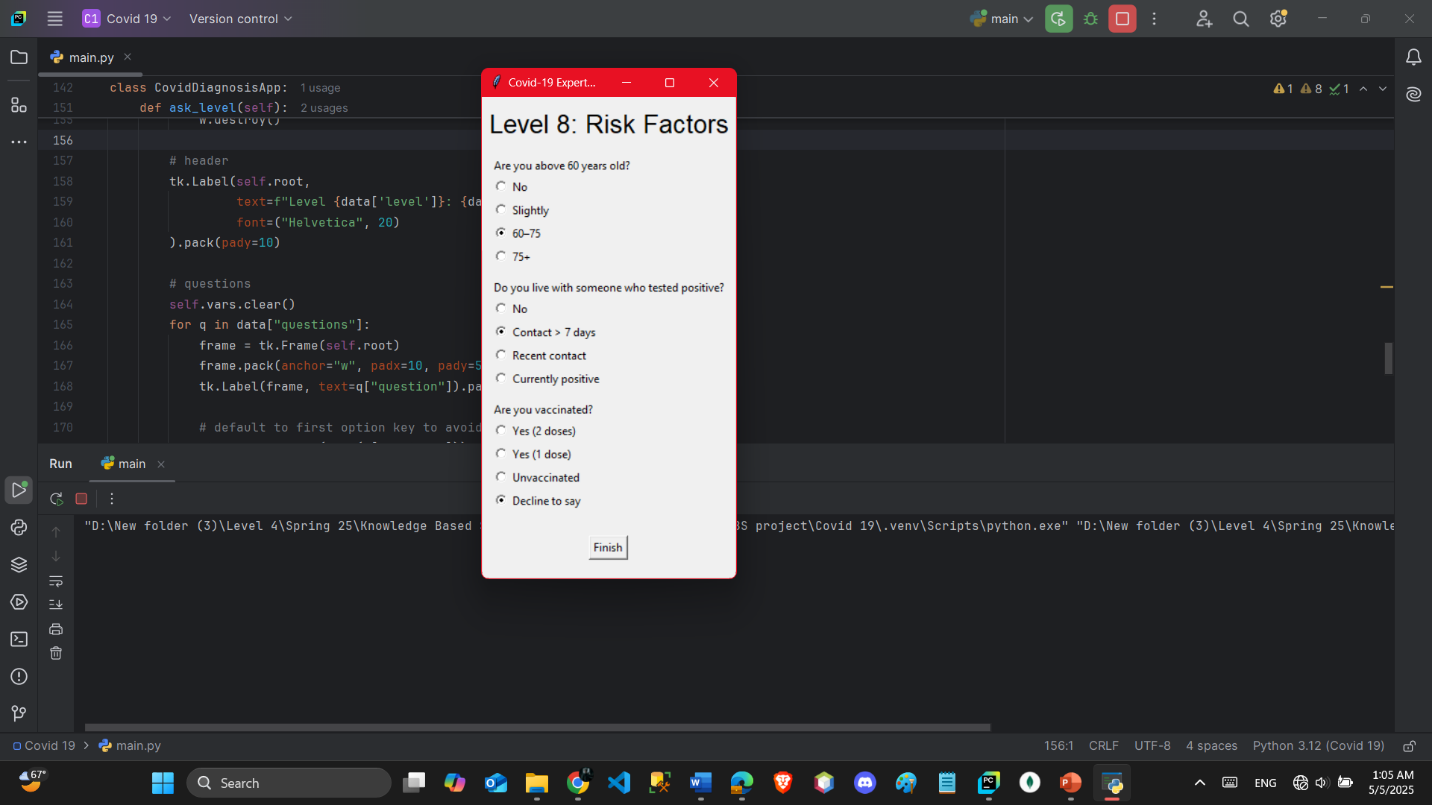
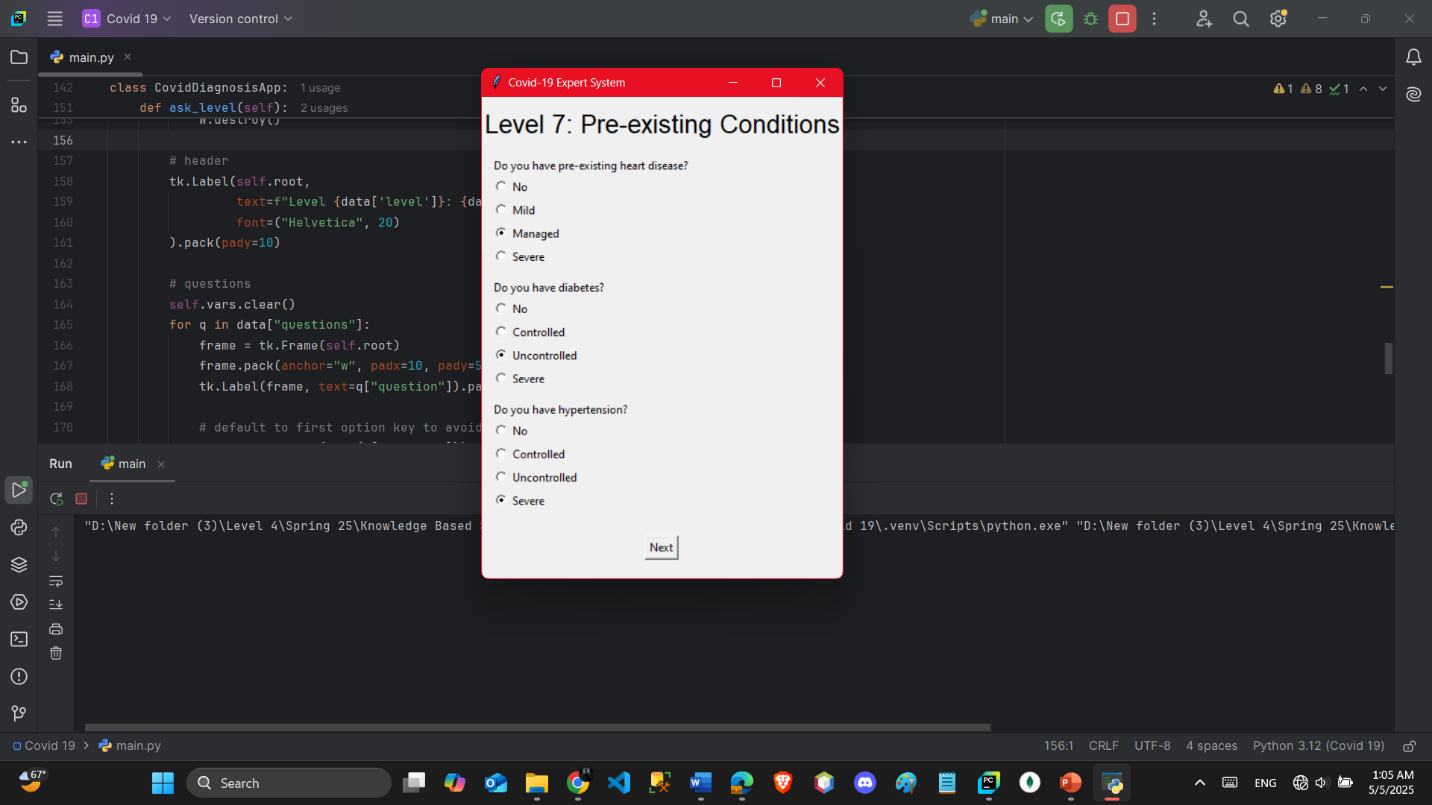
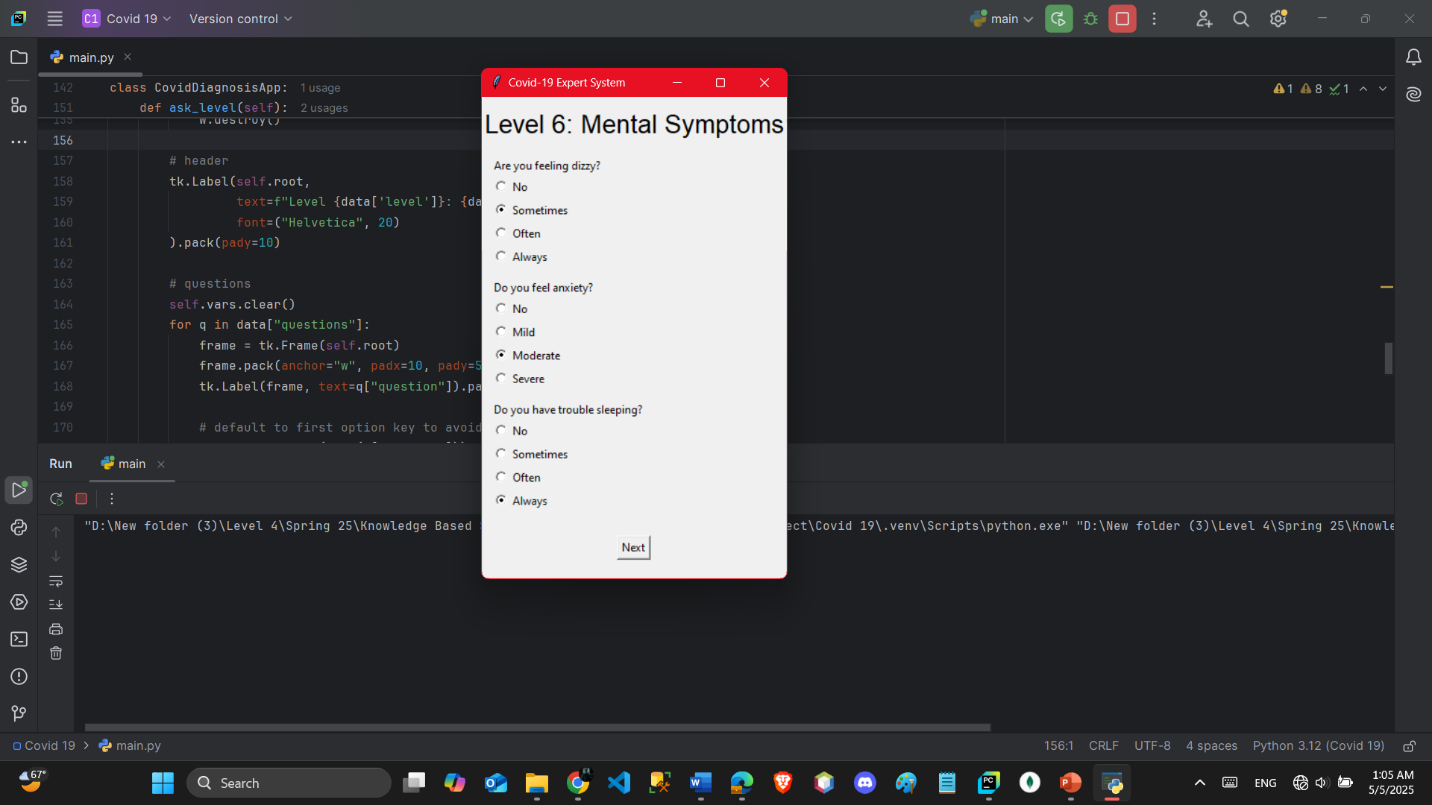
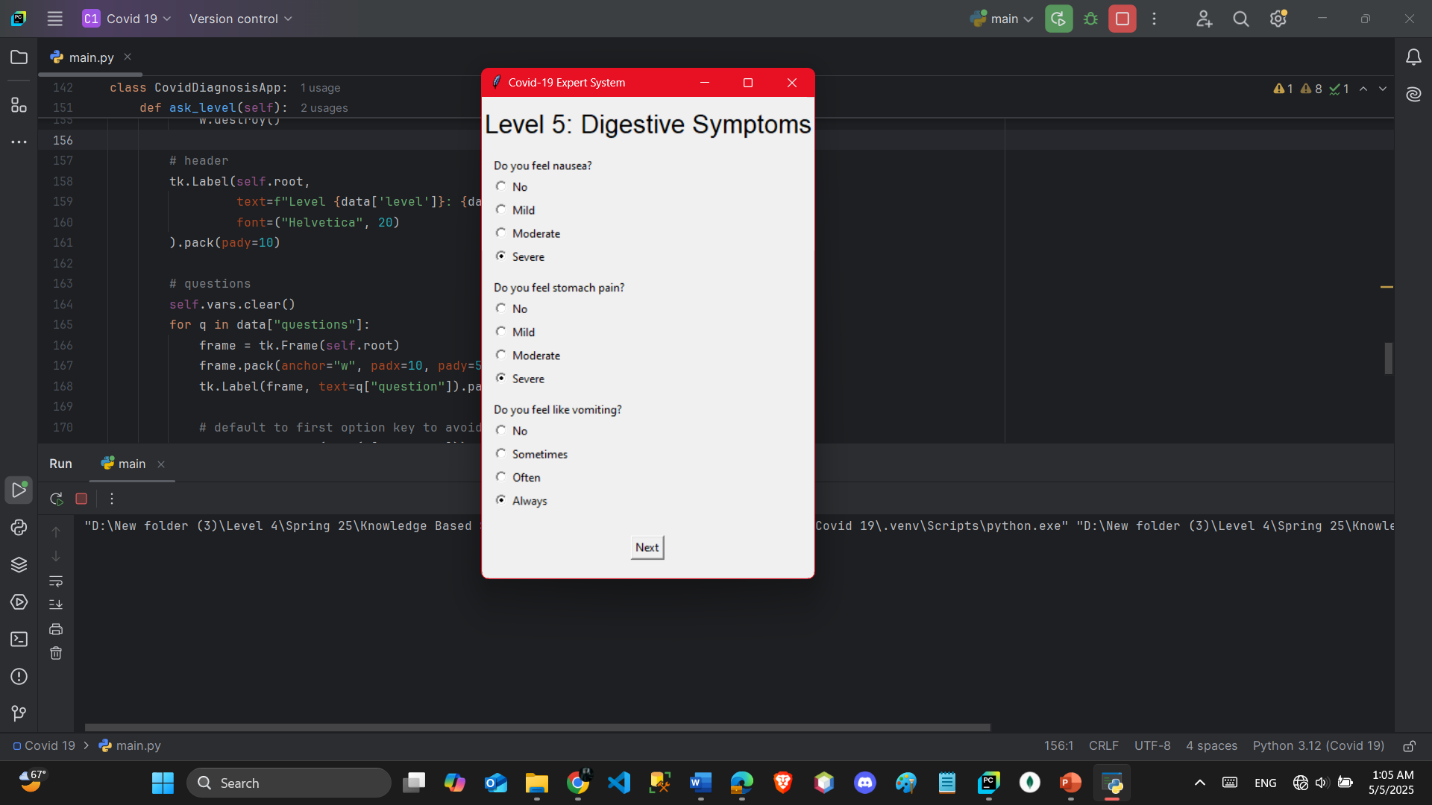
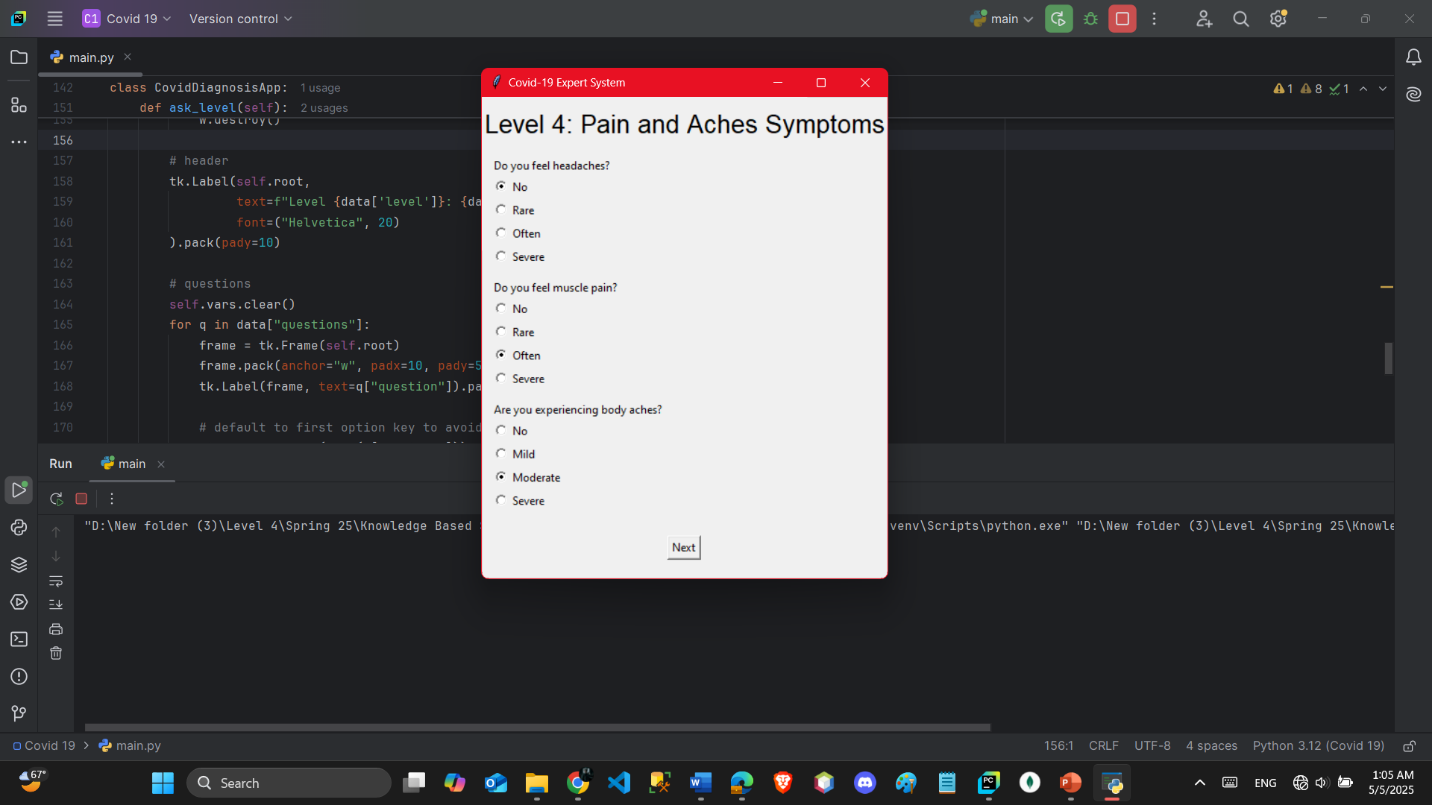
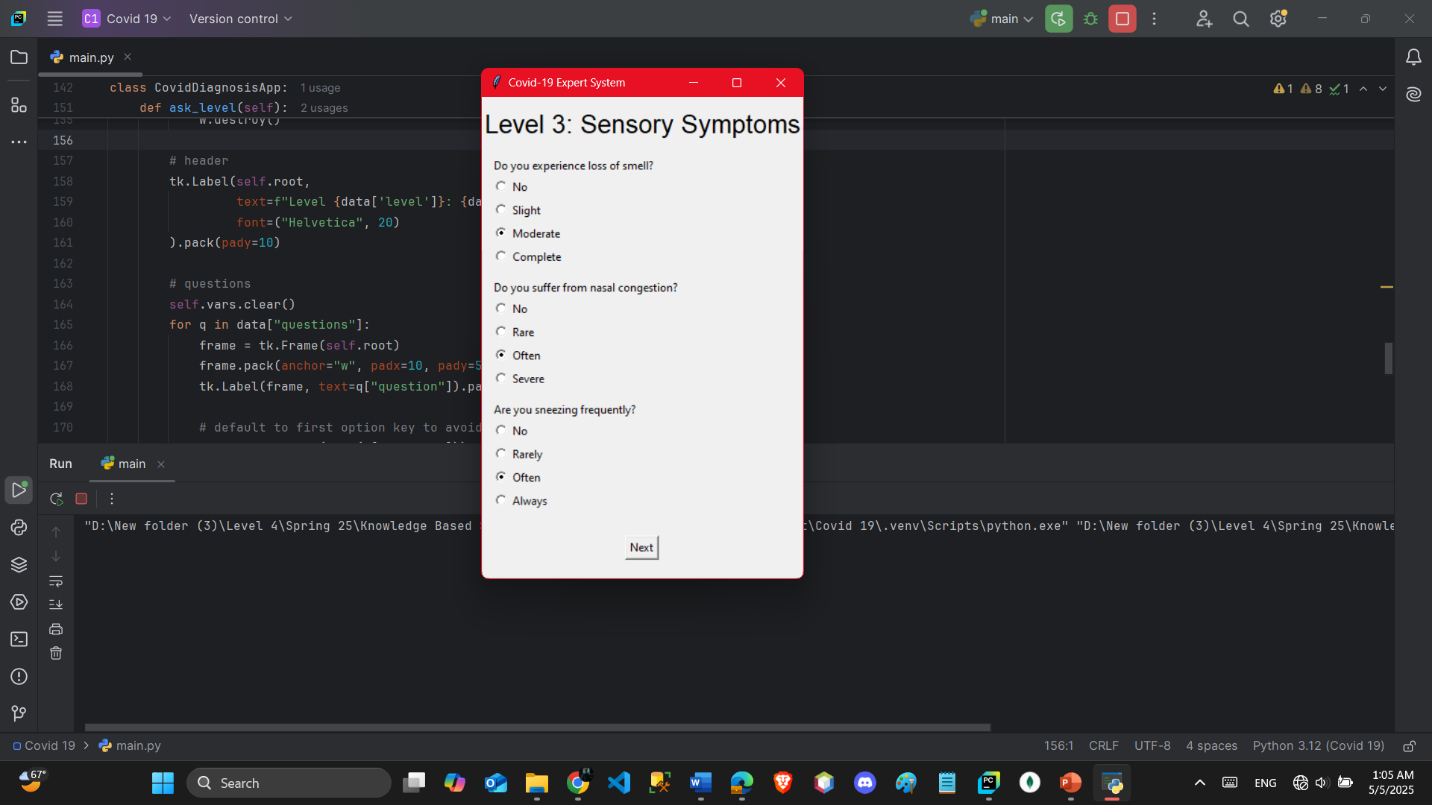
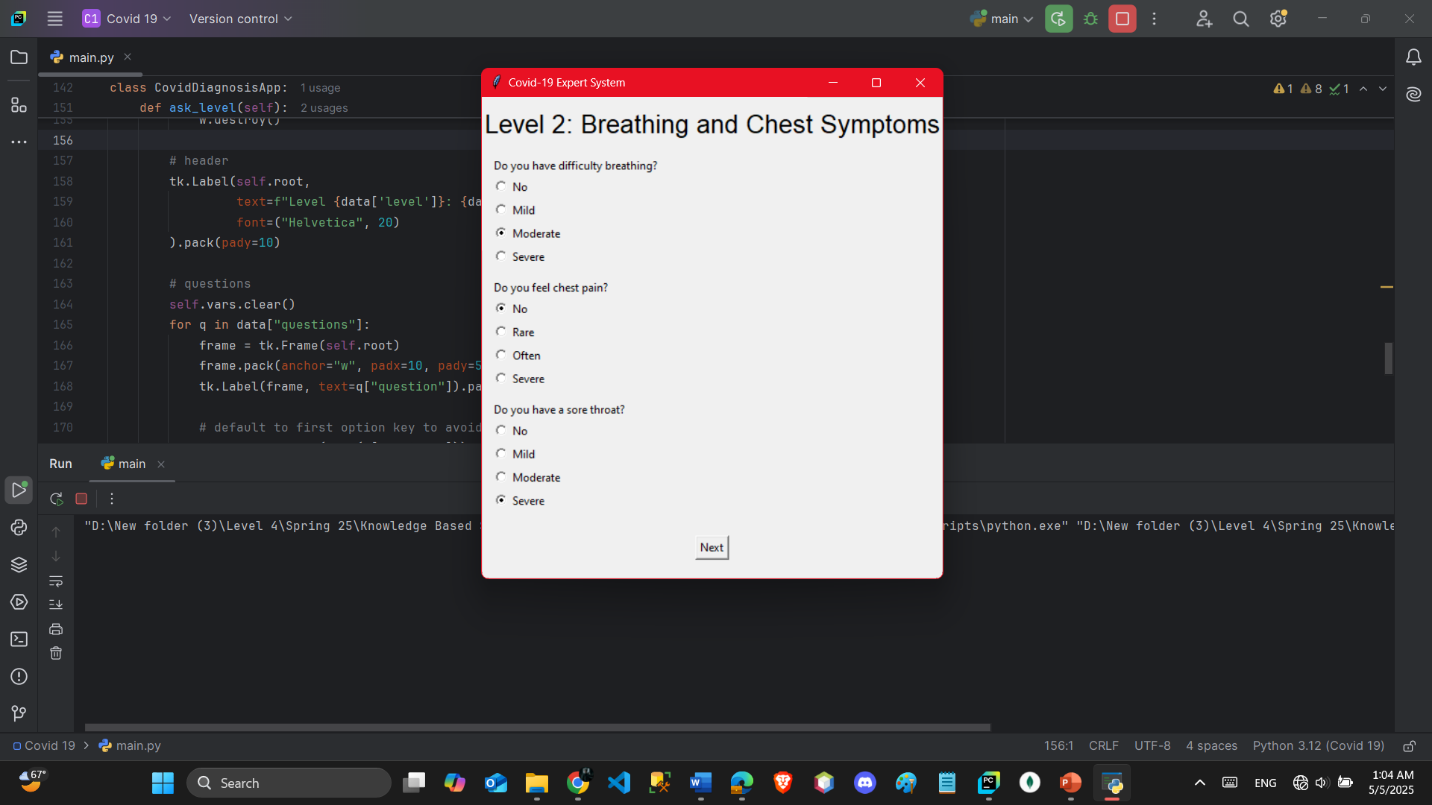
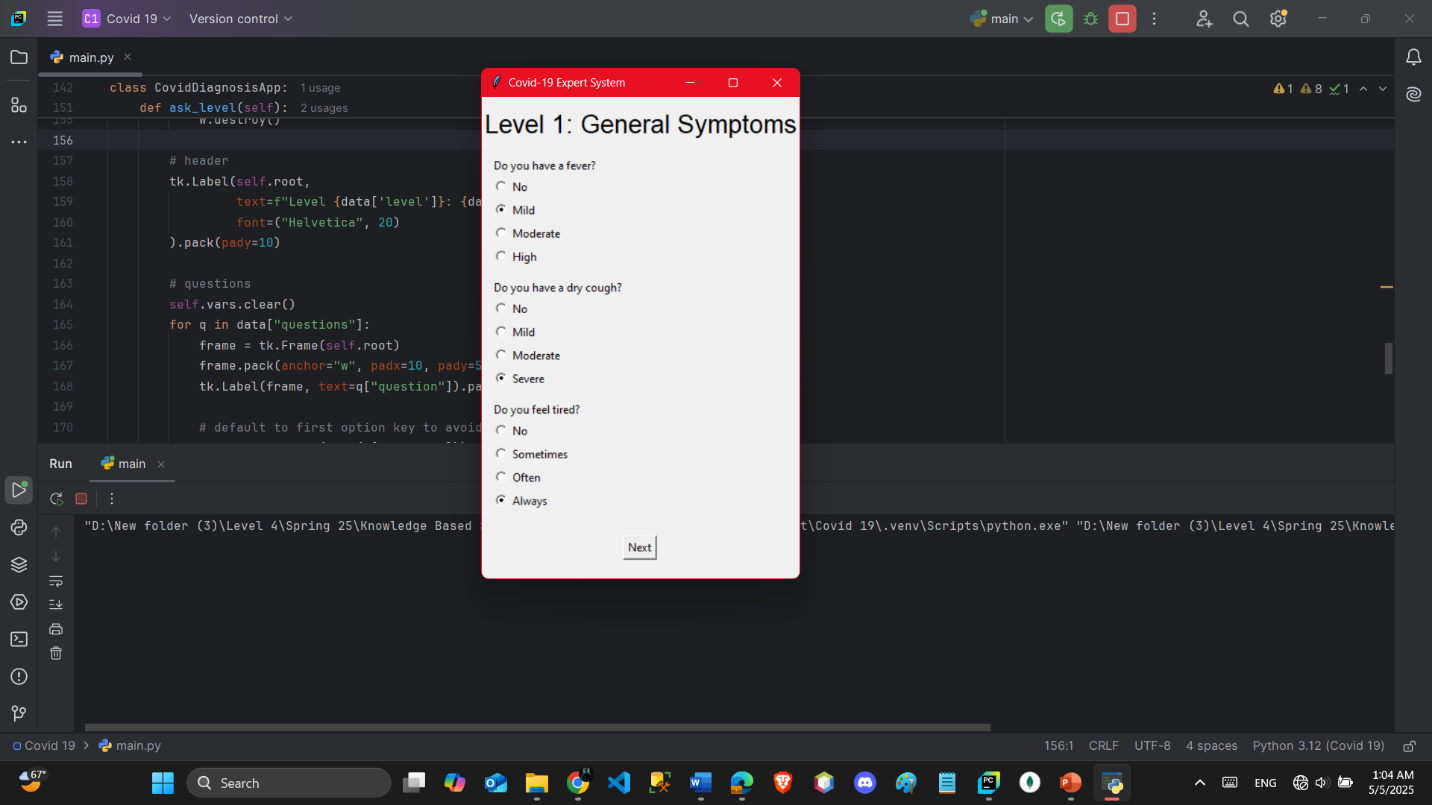
**Main Components:**

1. **COVID19ExpertSystem (KnowledgeEngine):** Core class where rules and facts are defined.
2. **Fact Definitions:** Each level collects facts via input prompts and passes them to the engine.
3. **Rule Base:** Logical if-then statements evaluate combinations of symptoms and risk factors.
4. **Risk Assessment:** Rules determine whether the user is at Low, Medium, or High risk.
5. **User Interface:** CLI-based question prompts, with colored output for recommendations (if supported).

**Sample Code Logic:**

**from experta import \*  
import tkinter as tk  
from tkinter import messagebox  
  
levels\_with\_questions = [  
 {  
 "level": 1,  
 "level\_title": "General Symptoms",  
 "questions": [  
 {"question": "Do you have a fever?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "High": 3}},  
 {"question": "Do you have a dry cough?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 {"question": "Do you feel tired?", "options": {"No": 0, "Sometimes": 1, "Often": 2, "Always": 3}},  
 # {"question": "Do you have chills?", "options": {"No": 0, "Rarely": 1, "Often": 2, "Always": 3}},  
 ]  
 },  
 {  
 "level": 2,  
 "level\_title": "Breathing and Chest Symptoms",  
 "questions": [  
 {"question": "Do you have difficulty breathing?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 {"question": "Do you feel chest pain?", "options": {"No": 0, "Rare": 1, "Often": 2, "Severe": 3}},  
 # {"question": "Is your voice hoarse?", "options": {"No": 0, "Slightly": 1, "Noticeable": 2, "Very Hoarse": 3}},  
 {"question": "Do you have a sore throat?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 ]  
 },  
 {  
 "level": 3,  
 "level\_title": "Sensory Symptoms",  
 "questions": [  
 {"question": "Do you experience loss of smell?", "options": {"No": 0, "Slight": 1, "Moderate": 2, "Complete": 3}},  
 # {"question": "Do you experience loss of taste?", "options": {"No": 0, "Slight": 1, "Moderate": 2, "Complete": 3}},  
 {"question": "Do you suffer from nasal congestion?", "options": {"No": 0, "Rare": 1, "Often": 2, "Severe": 3}},  
 {"question": "Are you sneezing frequently?", "options": {"No": 0, "Rarely": 1, "Often": 2, "Always": 3}},  
 ]  
 },  
 {  
 "level": 4,  
 "level\_title": "Pain and Aches Symptoms",  
 "questions": [  
 {"question": "Do you feel headaches?", "options": {"No": 0, "Rare": 1, "Often": 2, "Severe": 3}},  
 {"question": "Do you feel muscle pain?", "options": {"No": 0, "Rare": 1, "Often": 2, "Severe": 3}},  
 # {"question": "Are your joints aching?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 {"question": "Are you experiencing body aches?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 ]  
 },  
 {  
 "level": 5,  
 "level\_title": "Digestive Symptoms",  
 "questions": [  
 {"question": "Do you feel nausea?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 # {"question": "Do you have diarrhea?", "options": {"No": 0, "Mild": 1, "Frequent": 2, "Severe": 3}},  
 {"question": "Do you feel stomach pain?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 {"question": "Do you feel like vomiting?", "options": {"No": 0, "Sometimes": 1, "Often": 2, "Always": 3}},  
 ]  
 },  
 {  
 "level": 6,  
 "level\_title": "Mental Symptoms",  
 "questions": [  
 {"question": "Are you feeling dizzy?", "options": {"No": 0, "Sometimes": 1, "Often": 2, "Always": 3}},  
 # {"question": "Do you have confusion or brain fog?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 {"question": "Do you feel anxiety?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 {"question": "Do you have trouble sleeping?", "options": {"No": 0, "Sometimes": 1, "Often": 2, "Always": 3}},  
 ]  
 },  
 {  
 "level": 7,  
 "level\_title": "Pre-existing Conditions",  
 "questions": [  
 {"question": "Do you have pre-existing heart disease?", "options": {"No": 0, "Mild": 1, "Managed": 2, "Severe": 3}},  
 {"question": "Do you have diabetes?", "options": {"No": 0, "Controlled": 1, "Uncontrolled": 2, "Severe": 3}},  
 {"question": "Do you have hypertension?", "options": {"No": 0, "Controlled": 1, "Uncontrolled": 2, "Severe": 3}},  
 # {"question": "Do you have any autoimmune disease?", "options": {"No": 0, "Mild": 1, "Moderate": 2, "Severe": 3}},  
 ]  
 },  
 {  
 "level": 8,  
 "level\_title": "Risk Factors",  
 "questions": [  
 {"question": "Are you above 60 years old?", "options": {"No": 0, "Slightly": 1, "60–75": 2, "75+": 3}},  
 {"question": "Do you live with someone who tested positive?", "options": {"No": 0, "Contact > 7 days": 1, "Recent contact": 2, "Currently positive": 3}},  
 # {"question": "Do you work in a healthcare setting?", "options": {"No": 0, "Occasionally": 1, "Often": 2, "Always": 3}},  
 {"question": "Are you vaccinated?", "options": {"Yes (2 doses)": 0, "Yes (1 dose)": 1, "Unvaccinated": 2, "Decline to say": 3}},  
 ]  
 }  
]  
  
  
class CovidExpertSystem(KnowledgeEngine):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 self.risk\_result = None  
  
 @DefFacts()  
 def \_initial\_action(self):  
 yield Fact(action="diagnose")  
  
 # Low risk: total\_score ≤ 32 (about one-third of max 96)  
 @Rule(Fact(action='diagnose'),  
 Fact(total\_score=MATCH.s & P(lambda s: s <= 32)))  
 def low\_risk(self, s):  
 self.risk\_result = "Low Risk"  
  
 # Moderate risk: 33–64  
 @Rule(Fact(action='diagnose'),  
 Fact(total\_score=MATCH.s & P(lambda s: 33 <= s <= 64)))  
 def moderate\_risk(self, s):  
 self.risk\_result = "Moderate Risk"  
  
 # High risk: >64  
 @Rule(Fact(action='diagnose'),  
 Fact(total\_score=MATCH.s & P(lambda s: s > 64)))  
 def high\_risk(self, s):  
 self.risk\_result = "High Risk"  
  
  
class CovidExpertSystem(KnowledgeEngine):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 self.risk\_result = None  
  
 @DefFacts()  
 def \_initial\_action(self):  
 yield Fact(action="diagnose")  
  
 @Rule(Fact(action='diagnose'),  
 Fact(total\_score=MATCH.s & P(lambda s: s <= 32)))  
 def low\_risk(self, s):  
 self.risk\_result = "Low Risk"  
  
 @Rule(Fact(action='diagnose'),  
 Fact(total\_score=MATCH.s & P(lambda s: 33 <= s <= 64)))  
 def moderate\_risk(self, s):  
 self.risk\_result = "Moderate Risk"  
  
 @Rule(Fact(action='diagnose'),  
 Fact(total\_score=MATCH.s & P(lambda s: s > 64)))  
 def high\_risk(self, s):  
 self.risk\_result = "High Risk"  
  
  
class CovidDiagnosisApp:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 root.title("Covid-19 Expert System")  
 self.levels = len(levels\_with\_questions)  
 self.current\_level = 0  
 self.vars = []  
 self.ask\_level()  
  
 def ask\_level(self):  
 data = levels\_with\_questions[self.current\_level]  
 # clear previous widgets  
 for w in self.root.winfo\_children():  
 w.destroy()  
  
 # header  
 tk.Label(self.root,  
 text=f"Level {data['level']}: {data['level\_title']}",  
 font=("Helvetica", 16)  
 ).pack(pady=10)  
  
 # questions  
 self.vars.clear()  
 for q in data["questions"]:  
 frame = tk.Frame(self.root)  
 frame.pack(anchor="w", padx=10, pady=5)  
 tk.Label(frame, text=q["question"]).pack(anchor="w")  
  
 # default to first option key to avoid KeyError  
 first\_opt = next(iter(q["options"]))  
 var = tk.StringVar(value=first\_opt)  
 self.vars.append((var, q["options"]))  
  
 for opt\_label in q["options"]:  
 tk.Radiobutton(frame, text=opt\_label, variable=var, value=opt\_label).pack(anchor="w")  
  
 # next/finish button  
 btn\_text = "Next" if self.current\_level < self.levels - 1 else "Finish"  
 tk.Button(self.root, text=btn\_text, command=self.next\_level).pack(pady=20)  
  
 def next\_level(self):  
 # score this level  
 score = sum(options[var.get()] for var, options in self.vars)  
 setattr(self, f"level{self.current\_level+1}\_score", score)  
  
 self.current\_level += 1  
 if self.current\_level < self.levels:  
 self.ask\_level()  
 else:  
 self.evaluate()  
  
 def evaluate(self):  
 total = sum(getattr(self, f"level{i+1}\_score") for i in range(self.levels))  
 engine = CovidExpertSystem()  
 engine.reset()  
 engine.declare(Fact(total\_score=total))  
 engine.run()  
  
 risk = engine.risk\_result or "Unknown"  
 messagebox.showinfo("Diagnosis Result",  
 f"Total Score: {total}\nCovid-19 Risk: {risk}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 root = tk.Tk()  
 app = CovidDiagnosisApp(root)  
 root.mainloop()**

**RUN:**

****

# **Conclusion**

The COVID-19 Expert System demonstrates the effectiveness of rule-based systems in medical diagnostics. While it cannot replace real testing, it serves as a preliminary tool for assessing COVID-19 risk. It showcases the use of AI in healthcare, emphasizing transparency in decision-making via logical rules and structured questioning.

This project also strengthened our understanding of knowledge representation models, inference engines, and user interaction design.

# References

–, P. O. (n.d.). Retrieved from https://docs.python.org/

Documentation, E. L. (n.d.). *Experta Library Documentation*. Retrieved from Experta Library Documentation: Experta Library Documentation

Factors, C. –S. (n.d.).

Guidelines, W. H.–C.-1. (n.d.).

Norvig, A. I. (n.d.). Artificial Intelligence: A Modern Approach. In b. S. Norvig.