



សាកលវិទ្យាល័យនំរតុន
NORTON UNIVERSITY

**COLLAGE OF SCIENCES DEPARTMENT
OF COMPUTER STUDIE**

**YEAR 2 : GROUP ES1
SESSIO : EVENING**

Node Q

An AI-Powered Application for student Learning

Subject : Introduction Artificial Intelligence (AI)

LECTURER: Mr. Sek Sochea

OUR TEAM



Peng Maleap
ID : B20235253



Buor Synann
ID : B20242854

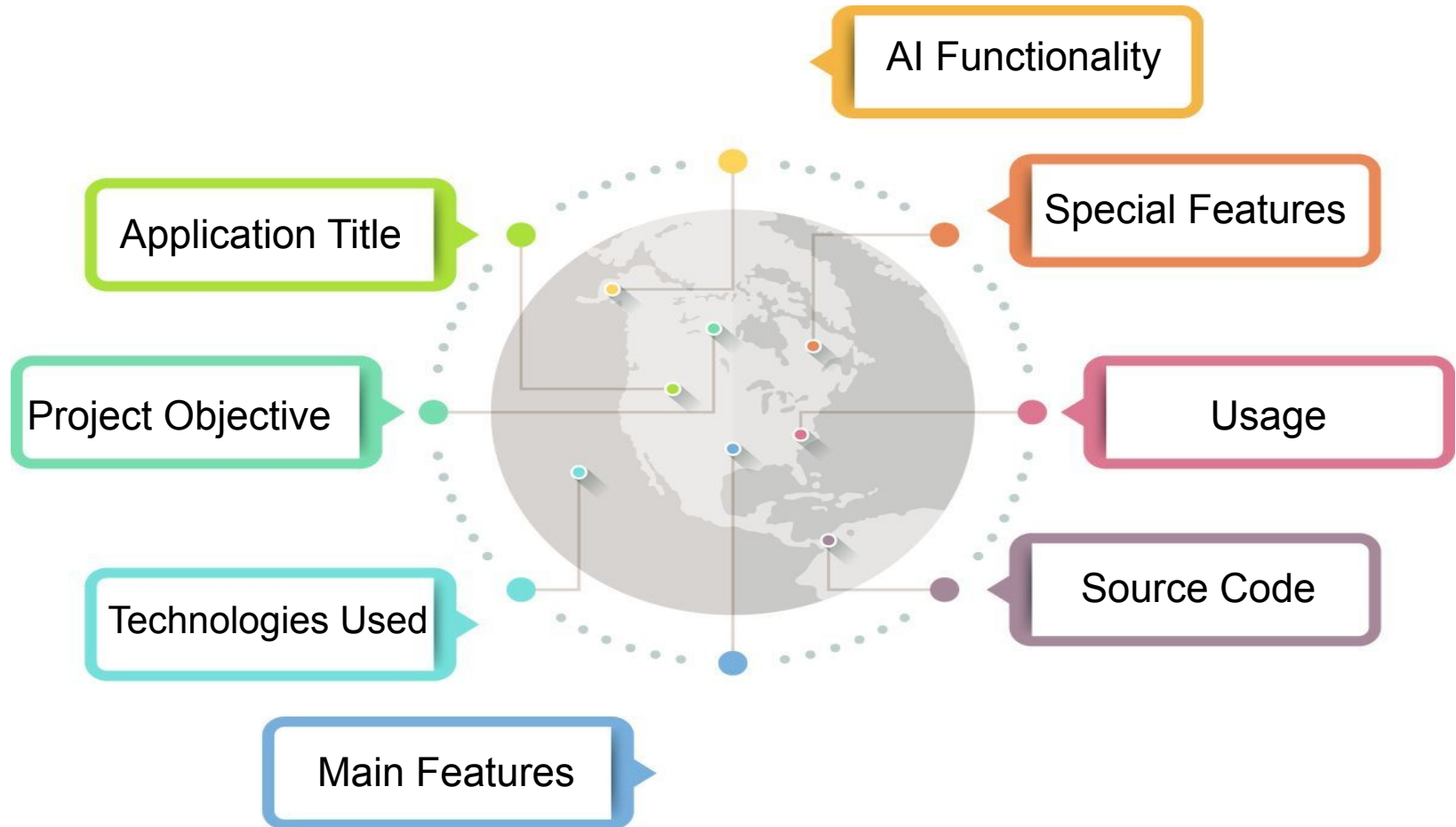


Ean Sokheng
ID : B20236655



Sang Mesa
ID : B20243361

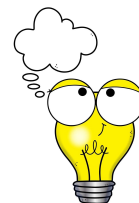
Table of Contents





ចំណងបើកម្មវិធី (Application Title)

NoteQ ជាកម្មវិធីអភិវឌ្ឍដោយបច្ចេកវិទ្យា AI សម្រាប់កុំព្យូទ័រ ដែលត្រូវបានរចនាឡើងដើម្បីជួយសិស្ស និងអ្នកសិក្សាឯករាជ្យក្នុងការបង្កើនមាតិកាបង្រៀនរបស់ពួកគេឲ្យ ក្លាយជាឧបករណ៍សិក្សាដែលមានប្រសិទ្ធភាព ។ មិនថាអ្នកវាយដោយដៃ អាប៊ីឡូតឯកសារ ឬនិយាយមាត់ក៏ ដើម្បីជួយឲ្យយល់បានងាយ សូត្រចាំបានល្អ និងអនុវត្តបានប្រសើរជាងមុន ។

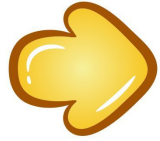




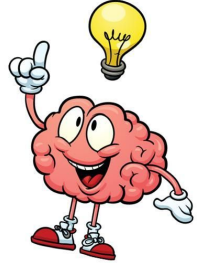
គោលបំណងនៃកម្មវិធី (Project Objective)

គម្រោងនេះមានបំណងក្នុងការបង្កើតកម្មវិធីមួយប្រើប្រាស់ភាសា **Python** និង **PyQt6** ដែលអាចឲ្យអ្នកប្រើបញ្ចូលចំណងជើងមេរៀន ឬ ឯកសារដែលផ្ទុកជាfile (PDF, Word, ឬ Text) វានឹងបង្កើតសំណួរ និងចម្លើយដោយស្វ័យប្រវត្តិ ដោយផ្អែកលើខ្លឹមសារមេរៀនដែលបានផ្តល់ ដោយប្រើបច្ចេកវិទ្យា AI ។ គោលដៅនៃកម្មវិធីគឺដើម្បីជួយសិស្សអាចរៀន ឬរៀបចំសំណួរត្រៀមប្រឡងបានងាយ ស្រួល និងរហ័ស ។ កម្មវិធីនេះនឹងជួយពង្រឹងការសិក្សានិងការអនុវត្តន៍បានល្អប្រសើរជាងមុនតាមរយៈការប្រើប្រាស់

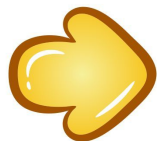




បច្ចេកវិទ្យាដែលប្រើ (Technologies Used)



- . **Backend Language** : Python3.10+
- . **GUI Framework** : PyQt6
- . **AI / NLP** : OpenAI API (GPT-4 or GPT-3.5),
or HuggingFace Transformers
- . **Optional Database** : SQLite or local JSON for Q&A
storage



មុខងារសំខាន់ៗនៃកម្មវិធី (Main Feature)



ល.រ	មុខងារ	ពិពណ៌នា
១	បញ្ចូលឯកសារ	ជ្រើសរើសឯកសារជា txt, pdf, ឬ docx ដើម្បីបញ្ជូនអោយ AI
២	បង្កើតសំណួរដោយ AI	ប្រើបច្ចេកវិទ្យា AI បង្កើតសំណួរដែលសមស្របទៅនឹងមាតិកា
៣	បង្ហាញ Q&A	បង្ហាញQ&Aលើផ្ទាំងកម្មវិធីដោយមានរចនាបទងាយស្រួលមើល
៤	រក្សាទុក Q&A	អ្នកប្រើអាចរក្សាទុកសំណួរនិងចម្លើយជាឯកសារ .txt ឬ .pdf
៥	ប្រវត្តិការប្រើប្រាស់	រក្សាទុកនូវសំណួរនិងចម្លើយដែលធ្លាប់បានបង្កើត
៧	លំហាត់តេស្ត	ធ្វើតេស្តសំណួរដែលបានបង្កើតនិងទទួលបានពិន្ទុដោយស្វ័យប្រវត្តិ



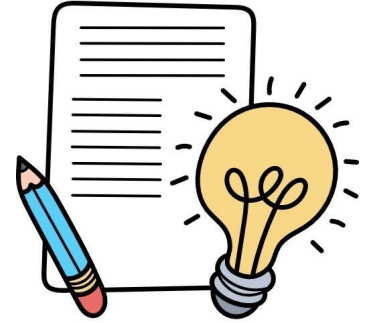
មុខងារ AI ដែលបានបញ្ចូល (AI Functionality)

- ◆ បម្លែងអត្ថបទជាសំណួរឬ ការបង្កើតសំណួរ (Text-to-Question Generation) :
AI នឹងប្រើ NLP (Natural Language Processing) ដើម្បីយល់អត្ថន័យនៃអត្ថបទឬឯកសារ
ហើយបង្កើតសំណួរដែលទាក់ទងនឹងមាតិកានោះ ។
- ◆ បង្កើតចម្លើយ (Answer Generation) : ប្រសិនបើអ្នកប្រើចង់បានចម្លើយផងដែរ AI
នឹងបញ្ចេញចម្លើយត្រឹមត្រូវសម្រាប់សំណួរដែលបានបង្កើត ។





លក្ខណៈពិសេសនៃកម្មវិធី (Special Features)



- **បញ្ចូលអត្ថបទឬឯកសារ (Text or File Upload):**
អ្នកប្រើអាចបញ្ចូលពាក្យ ឬឯកសារដូចជា PDF, DOCX, TXT ។
- **បង្កើតសំណួរចំនួន ១០ (Generate 10 Questions):**
កម្មវិធីនឹងបង្កើតសំណួរចំនួន ១០ ដែលមានទំនាក់ទំនងជាមួយមាតិកា ។
- **បង្ហាញចម្លើយនៃសំណួរ (Optional Answers):**
អ្នកអាចទទួលបានចម្លើយពីការបង្កើតសំណួរ ។
- **អាចរក្សាទុកឯកសារ Q&A (Export/Save Q&A):**
អ្នកប្រើអាចបញ្ចេញ Q&A ជា PDF ឬរក្សាទុកក្នុងប្រព័ន្ធ ។



របៀបប្រើប្រាស់ (Usage)

ជំហានទី ១ ៖ ការចូលគណនី

ដំបូង

អ្នកប្រើប្រាស់ត្រូវតែចូលទៅក្នុងកម្មវិធី ។ ពួកគេអាចធ្វើ

ដូច្នេះបានដោយប្រើឈ្មោះអ្នកប្រើ

និងពាក្យសម្ងាត់ ឬ

ជ្រើសរើសជម្រើសចូលរហ័សតាមរយៈគណនី Facebook

ឬ Google

Login

Hi, Welcome! 🙌

Username:

Enter username

Password:

Enter password

Log in

Register

Ready to Learn? Choose Your Path!

Node Q

Your AI Assistant

Use AI to study smarter. Generate questions from your lesson notes and keep important points organized for quick review.

Quizzzy

Your Study Buddy

Practice what you've learned. Get auto-generated exercises to help you prepare for exams and track your progress.

ជំហានទី ២ ៖ ការជ្រើសរើសម៉ូឌុល

បន្ទាប់ពីចូលបានជោគជ័យ អ្នកប្រើប្រាស់នឹងត្រូវបានបង្ហាញ ជាមួយនឹងអេក្រង់ជ្រើសរើស។ ពួកគេអាច

Node Q: សម្រាប់បង្កើតសំណួរ និងរៀបចំកំណត់ត្រា ។

Quizzzy: សម្រាប់អនុវត្ត និងធ្វើតេស្តចំណេះដឹង ។

ជំហានទី ៣៖ ការប្រើប្រាស់ Node Q (ជំនួយការ AI)

ដើម្បីចាប់ផ្តើម អ្នកប្រើប្រាស់នឹងជ្រើសរើស Node Q ។
ពួកគេអាចបញ្ចូលខ្លឹមសារមេរៀនរបស់ពួកគេតាមពីរវិធី ៖

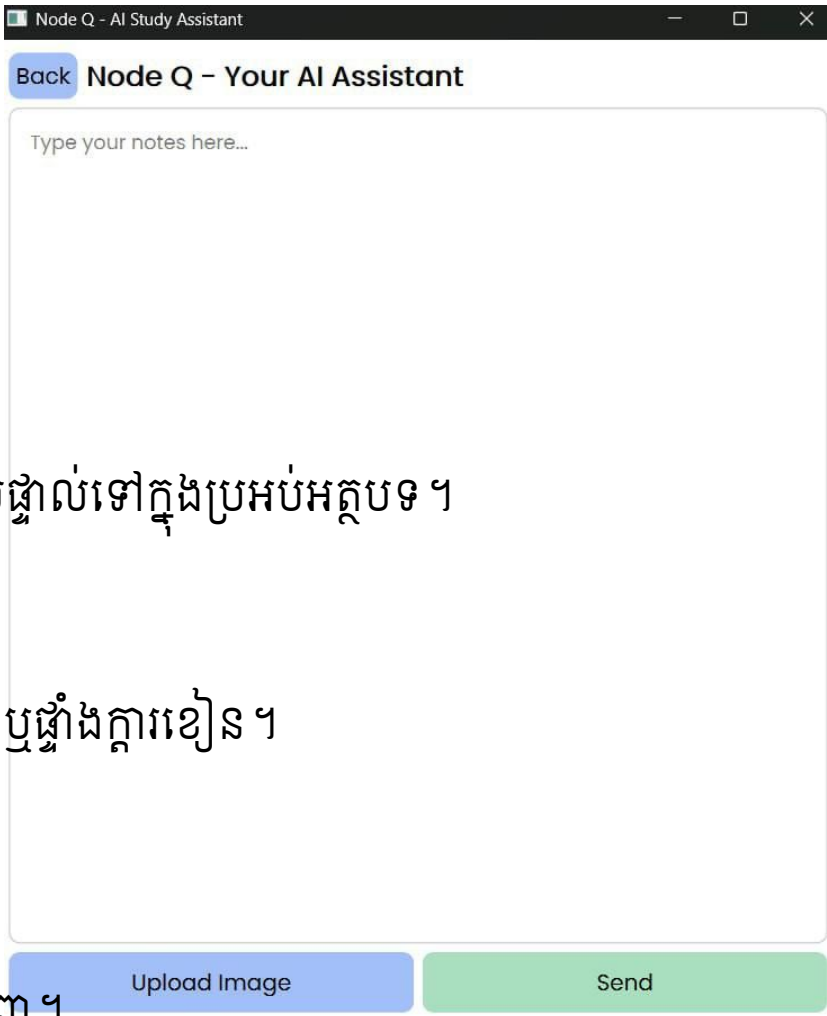
វាយបញ្ចូលអត្ថបទ ៖
បិទ

ក្លាប់អត្ថបទពីកំណត់ត្រា ឬវាយ បញ្ចូលគំនិតរបស់ពួកគេដោយផ្ទាល់ទៅក្នុងប្រអប់អត្ថបទ ។

បញ្ចូលរូបភាព ៖ ចុចប៊ូតុង “Upload Image” ដើម្បី
ប

ញ្ចូលរូបភាពកំណត់ត្រាដែលសរសេរដោយដៃ រូបថត សៀវភៅ ឬផ្ទាំងក្តារខៀន ។

បន្ទាប់ពីបញ្ចូលខ្លឹមសាររួចហើយ អ្នកប្រើប្រាស់ចុចប៊ូតុង
"Send" ។ AI របស់ Node Q នឹងដំណើរការខ្លឹមសារនោះ
ហើយបង្កើតសំណួរ
ផ្ទាល់ខ្លួន ក៏ដូចជារៀបចំចំណុចសំខាន់ៗ សម្រាប់ពិនិត្យឡើងវិញ ។





Back

Quiz – PyQt6 + SQLite

Q1: What is the capital of France?

Berlin

Madrid

Paris

Rome

Submit

ជំហានទី ៤ ៖ ការប្រើប្រាស់ Quizzzy (អ្នកគ្រប់គ្រងគម្រោង)

នៅពេលដែល Node Q
បានបង្កើតសម្ភារៈសិក្សារួចហើយ
អ្នកប្រើប្រាស់អាចប្តូរទៅកាន់ម៉ូឌុល Quizzzy
ដើម្បីអនុវត្ត ។

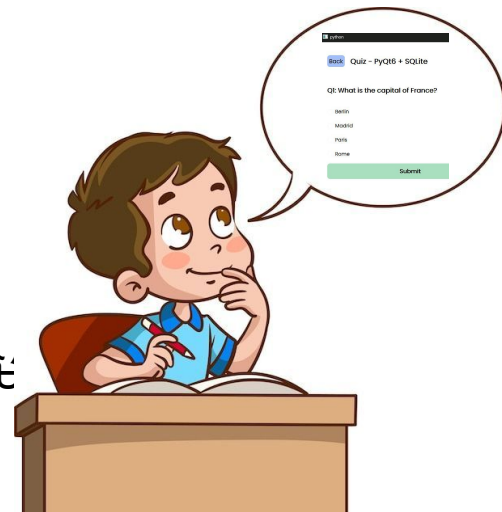
Quizzzy នឹងបង្កើតលំហាត់
និងសំណួរដោយស្វ័យប្រវត្តិដោយ
ផ្អែកលើកំណត់ត្រាដែលពួកគេបានបញ្ចូលក្នុង
Node Q1

អ្នកប្រើប្រាស់អាចចាប់ផ្តើមអនុវត្ត
និងឆ្លើយសំណួរទាំងនោះ ហើយ Quizzzy
នឹងជួយពួកគេតាមដានពី
និងវឌ្ឍនភាពរបស់
ពួកគេដើម្បីមើលថាពួកគេត្រូវការបង្កើនការយកចិត្ត
កលើ ផ្នែកណាខ្លះ ។



ជំហានទី ៥ ៖ ការពិនិត្យឡើងវិញ និងធ្វើម្តងទៀត

ដំណើរការនេះអាចធ្វើឡើងវិញបានច្រើនដងតាមដែលត្រូវការ ។
អ្នកប្រើប្រាស់អាចបញ្ចូលមេរៀនថ្មីៗទៅក្នុង Node Q ហើយ បន្ទាប់មក
Quizzy ដើម្បីអនុវត្ត ។



វដ្តនៃការបញ្ចូលមេរៀន (Node Q) និងការអនុវត្តសកម្ម (Quizzy)
នេះជួយឱ្យពួកគេចងចាំមេរៀនបានល្អប្រសើរ និង
ត្រៀមខ្លួនសម្រាប់ការប្រឡងបានយ៉ាងងាយស្រួល ។

Source Code

```
1 # Import necessary PyQt6 modules for UI components and functionality
2 from PyQt6.QtWidgets import QWidget, QLabel, QLineEdit, QPushButton, QVBoxLayout, QMessageBox, QCheckBox
3 from db.database import Database # Your custom database class for user validation
4 from ui.main_window import MainWindow # The main application window after login
5 from ui.register_form import RegisterForm # The registration form window
6 from PyQt6.QtCore import Qt # Qt constants (not directly used here but imported)
7
8 # Define the LoginForm class which is a QWidget (a basic window/container)
9 class LoginForm(QWidget):
10     def __init__(self):
11         super().__init__() # Call the constructor of QWidget
12
13         # Set the style for this widget (background color)
14         self.setStyleSheet("""
15             LoginForm {
16                 background-color: white;
17             }
18         """)
19
20         self.setWindowTitle("Login") # Set the window title
21         self.resize(556, 691) # Set the size of the login window
22
23         self.db = Database() # Create an instance of your custom Database class
24         self.setup_ui() # Call method to setup the UI elements
25
26     def setup_ui(self):
27         # Use a vertical layout to arrange widgets from top to bottom
28         layout = QVBoxLayout()
29         layout.setContentsMargins(30, 30, 30, 30) # Add padding around the edges
30         layout.setSpacing(15) # Space between widgets
31
32         # Greeting Label
33         self.title_label = QLabel("Hi, Welcome! 🌟")
34         self.title_label.setText("""
35             <span style='font-size: 30px; font-weight: 700; color: black;'>Hi, Welcome! 🌟</span>
36         """)
37         layout.addWidget(self.title_label) # Add to layout
38
39         # Username Label and input field
40         self.label_username = QLabel("Username")
41         self.input_username = QLineEdit()
42         self.input_username.setStyleSheet("padding: 20px; background-color: white; border: 1px solid #000000; color: black; border-radius: 12px;")
43         self.input_username.setPlaceholderText("Enter username") # Hint text
44         layout.addWidget(self.label_username)
45         layout.addWidget(self.input_username)
46
47         # Password Label and input field
48         self.label_password = QLabel("Password")
49         self.input_password = QLineEdit()
50         self.input_password.setStyleSheet("padding: 20px; background-color: white; border: 1px solid #000000; color: black; border-radius: 12px; margin-bottom: 20px;")
51         self.input_password.setPlaceholderText("Enter password")
52         self.input_password.setEchoMode(QLineEdit.EchoMode.Password) # Hide password characters
53         layout.addWidget(self.label_password)
54         layout.addWidget(self.input_password)
```

```
55
56 # Login button
57 self.login_button = QPushButton("Log in")
58 self.login_button.setStyleSheet("""
59     background-color: black;
60     color: white;
61     padding: 10px;
62     border: none;
63     border-radius: 5px;
64 """)
65 self.login_button.clicked.connect(self.check_login) # Call check_login when clicked
66 layout.addWidget(self.login_button)
67
68 # Register button
69 self.register_button = QPushButton("Register")
70 self.register_button.setStyleSheet("""
71     background-color: black;
72     color: white;
73     padding: 10px;
74     border: none;
75     border-radius: 5px;
76 """)
77 self.register_button.clicked.connect(self.go_to_register) # Go to register form when clicked
78 layout.addWidget(self.register_button)
79
80 self.setLayout(layout) # Set the layout for the widget
81
82 def check_login(self):
83     # Get the entered username and password
84     username = self.input_username.text()
85     password = self.input_password.text()
86
87     # Validate the credentials using your Database class
88     user = self.db.validate_user(username, password)
89
90     if user:
91         # If valid, show success message and open the main app window
92         QMessageBox.information(self, "Success", "Login successful!")
93         self.open_main_window()
94     else:
95         # If invalid, show error message
96         QMessageBox.warning(self, "Error", "Invalid username or password.")
97
98 def go_to_register(self):
99     # Open the RegisterForm window and close the Login form
100     self.main_window = RegisterForm()
101     self.main_window.show()
102     self.close()
103
104 def open_main_window(self):
105     # Open the MainWindow (main application screen) and close the Login form
106     self.main_window = MainWindow()
107     self.main_window.show()
108     self.close()
109
```

```

1 from PyQt6.QtWidgets import (
2     QMainWindow, QLabel, QWidget, QVBoxLayout, QHBoxLayout,
3     QGraphicsDropShadowEffect, QPushButton
4 )
5 from PyQt6.QtCore import Qt
6 from PyQt6.QtGui import QPixmap, QColor
7 from ui.quizzy import QuizWindow
8 from ui.notes_ai import NodeQWindow # Import the NodeQWindow Class
9
10 class CardWidget(QWidget):
11     def __init__(self, icon_path, title, subtitle, description, button_color):
12         super().__init__()
13         self.setStyleSheet("""
14             QWidget {
15                 background-color: transparent;
16             }
17             """)
18
19         card_container = QWidget()
20         card_container.setStyleSheet("""
21             QWidget {
22                 background-color: #f5f5f5;
23                 border-radius: 15px;
24                 padding: 20px;
25             }
26             """)
27
28         shadow = QGraphicsDropShadowEffect()
29         shadow.setBlurRadius(15)
30         shadow.setOffset(0)
31         shadow.setOffset(5)
32         shadow.setColor(QColor("#0000001A"))
33         card_container.setGraphicsEffect(shadow)
34
35         layout = QVBoxLayout()
36         layout.setSpacing(15)
37         layout.setContentsMargins(0, 0, 0, 0)
38
39         icon_container = QLabel()
40         icon_container.setFixedSize(40, 40)
41         icon_container.setStyleSheet("""
42             QWidget {
43                 background-color: {button_color};
44                 border-radius: 20px;
45                 padding: 5px;
46             }
47             """)
48
49         pixmap = QPixmap(icon_path).scaled(
50             24, 24, Qt.AspectRatioMode.KeepAspectRatio, Qt.TransformationMode.SmoothTransformation
51         )
52         icon_container.setPixmap(pixmap)
53         icon_container.setAlignment(Qt.AlignmentFlag.AlignCenter)
54         layout.addWidget(icon_container)
55
56         text_layout = QVBoxLayout()
57         text_layout.setSpacing(5)
58
59         title_label = QLabel(title)
60         title_label.setStyleSheet("font-size: 18px; font-weight: bold; color: #333;")
61         text_layout.addWidget(title_label)
62
63         subtitle_label = QLabel(subtitle)
64         subtitle_label.setStyleSheet("font-size: 12px; color: #666;")
65         text_layout.addWidget(subtitle_label)
66
67         desc_label = QLabel(description)
68         desc_label.setStyleSheet("font-size: 14px; color: #666;")
69         desc_label.setWordWrap(True)
70         text_layout.addWidget(desc_label)
71
72         layout.addLayout(text_layout)
73
74         arrow_button = QPushButton()
75         arrow_button.setFixedSize(30, 30)
76         arrow_button.setStyleSheet("""
77             QPushButton {
78                 background-color: {button_color};
79                 border: none;
80                 border-radius: 15px;
81             }
82             """)
83         layout.addWidget(arrow_button)

```

```

78
79         card_container_layout = QHBoxLayout(card_container)
80         card_container_layout.addLayout(layout)
81
82         main_layout = QVBoxLayout(self)
83         main_layout.addWidget(card_container)
84
85
86 class MainWindow(QMainWindow):
87     def __init__(self):
88         super().__init__()
89         self.setStyleSheet("""
90             MainWindow {
91                 background-color: white;
92             }
93             """)
94         self.setWindowTitle("Node Q")
95         self.resize(556, 691)
96
97         central_widget = QWidget()
98         main_layout = QVBoxLayout(central_widget)
99         main_layout.setAlignment(Qt.AlignmentFlag.AlignTop)
100         main_layout.setSpacing(30)
101         main_layout.setContentsMargins(20, 50, 20, 20)
102
103         title_label = QLabel("Ready to Learn? Choose Your Path!")
104         title_label.setStyleSheet("font-size: 30px; font-weight: 700; color: black;")
105         main_layout.addWidget(title_label)
106
107         node_q_card = CardWidget(
108             "image/mingcute_ai-fill.png",
109             "Node Q",
110             "Your AI Assistant",
111             "Use AI to study smarter. Generate questions from your lesson notes and keep important points organized for quick review.",
112             "#a3bffa" # Light purple
113         )
114         node_q_card.mousePressEvent = lambda event: self.open_node_q_window()
115         main_layout.addWidget(node_q_card)
116
117         quizzy_card = CardWidget(
118             "image/material-symbols_quiz-rounded.png",
119             "Quizzy",
120             "Project Manager",
121             "Practice what you've learned. Get auto-generated exercises to help you prepare for exams and track your progress.",
122             "#a9d9fb" # Light green
123         )
124         quizzy_card.mousePressEvent = lambda event: self.open_quiz()
125         main_layout.addWidget(quizzy_card)
126
127         self.setCentralWidget(central_widget)
128
129     def open_node_q_window(self):
130         self.node_q_window = NodeQWindow(main_window=self)
131         self.node_q_window.show()
132         self.close()
133     def open_quiz(self):
134         self.quiz_window = QuizWindow(main_window=self)
135         self.quiz_window.show()
136         self.hide()
137

```



```

1 from PyQt6.QtWidgets import (
2     QMainWindow, QWidget, QTextEdit, QPushButton,
3     QVBoxLayout, QHBoxLayout, QFileDialog, QLabel
4 )
5
6 class NodeQWindow(QMainWindow):
7     def __init__(self, main_window=None):
8         super().__init__()
9         self.main_window = main_window # store reference to main window
10
11         self.setWindowTitle("Node Q - AI Study Assistant")
12         self.resize(600, 700)
13         self.setStyleSheet("background-color: white;")
14
15         # Central widget and layout
16         central_widget = QWidget()
17         self.setCentralWidget(central_widget)
18         main_layout = QVBoxLayout(central_widget)
19
20         # Header Layout (Back button + Title)
21         header_layout = QHBoxLayout()
22         back_btn = QPushButton("Back")
23         back_btn.setStyleSheet(
24             """
25             QPushButton {
26                 "background-color: #a3bffa;"
27                 "color: black;"
28                 "padding: 5px;"
29                 "border-radius: 8px;"
30                 "} QPushButton:hover {"
31                 "background-color: #dfefff;"
32                 "}"
33             """
34         )
35         back_btn.clicked.connect(self.goto_main)
36         header_layout.addWidget(back_btn)
37
38         header = QLabel("Node Q - Your AI Assistant")
39         header.setStyleSheet("font-size: 20px; font-weight: bold; color: black;")
40         header_layout.addWidget(header)
41         header_layout.addStretch()
42         main_layout.addLayout(header_layout)
43
44         # Text input area
45         self.text_input = QTextEdit()
46         self.text_input.setPlaceholderText("Type your notes here...")
47         self.text_input.setStyleSheet("""
48             QTextEdit {
49                 font-size: 14px;
50                 padding: 10px;
51                 border: 1px solid #ccc;
52                 border-radius: 8px;
53                 color: black;
54             }
55             """)
56         main_layout.addWidget(self.text_input, stretch=1)

```

```

56         # Upload & Send buttons
57         button_layout = QHBoxLayout()
58         upload_btn = QPushButton("Upload Image")
59         upload_btn.setStyleSheet(
60             """
61             QPushButton {
62                 "background-color: #a3bffa;"
63                 "color: black;"
64                 "padding: 10px 16px;"
65                 "border: none;"
66                 "border-radius: 8px;"
67                 "} QPushButton:hover {"
68                 "background-color: #dfefff;"
69                 "}"
70             """
71         )
72         upload_btn.clicked.connect(self.upload_image)
73         button_layout.addWidget(upload_btn)
74
75         send_btn = QPushButton("Send")
76         send_btn.setStyleSheet(
77             """
78             QPushButton {
79                 "background-color: #a9dffb;"
80                 "color: black;"
81                 "padding: 10px 16px;"
82                 "border: none;"
83                 "border-radius: 8px;"
84                 "} QPushButton:hover {"
85                 "background-color: #dfefff;"
86                 "}"
87             """
88         )
89         send_btn.clicked.connect(self.send_text)
90         button_layout.addWidget(send_btn)
91
92         main_layout.addLayout(button_layout)
93
94         def upload_image(self):
95             file_path, _ = QFileDialog.getOpenFileName(
96                 self, "Upload Image", "", "Images (*.png *.jpeg *.bmp)")
97             if file_path:
98                 print(f"📁 Uploaded Image: {file_path}")
99
100         def send_text(self):
101             user_text = self.text_input.toPlainText().strip()
102             if user_text:
103                 print(f"💬 User Message: {user_text}")
104                 self.text_input.clear()
105
106         def goto_main(self):
107             if self.main_window:
108                 self.main_window.show()
109             self.close()

```

```

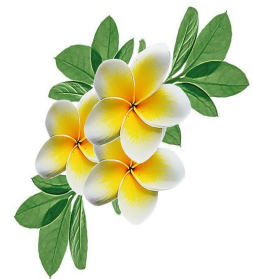
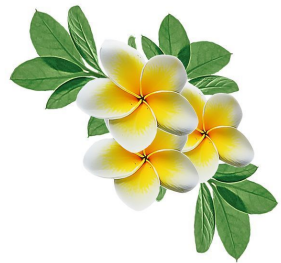
1 from PyQt6.QtWidgets import (
2     QMainWindow, QWidget, QVBoxLayout, QHBoxLayout,
3     QLabel, QPushButton, QRadioButton, QMessageBox, QButtonGroup
4 )
5 from PyQt6.QtCore import Qt
6
7
8 class QuizWindow(QMainWindow):
9     def __init__(self, main_window=None):
10         super().__init__()
11         self.main_window = main_window # Reference to MainWindow
12
13         self.resize(500, 400)
14         self.setStyleSheet("background-color: white;")
15
16         # Connect to DB and get questions
17         from db.database import Database # import here to avoid circular issues
18         self.db = Database()
19         self.questions_data = self.db.get_all_questions()
20         self.current_index = 0
21
22         if not self.questions_data:
23             QMessageBox.critical(self, "Error", "No questions found in the database.")
24             self.close()
25             return
26
27         central_widget = QWidget()
28         self.setCentralWidget(central_widget)
29         self.main_layout = QVBoxLayout(central_widget)
30         self.main_layout.setContentsMargins(20, 20, 20, 20)
31         self.main_layout.setSpacing(15)
32
33         # --- Back button and header layout ---
34         header_layout = QHBoxLayout()
35         back_btn = QPushButton("Back")
36         back_btn.setStyleSheet(
37             """
38             QPushButton {
39                 background-color: #a3bffa;
40                 color: black;
41                 padding: 5px;
42                 border-radius: 8px;
43             }
44             QPushButton:hover {
45                 background-color: #dfefff;
46             }
47             """
48         )
49         back_btn.clicked.connect(self.goto_main)
50         header_layout.addWidget(back_btn)
51
52         title_label = QLabel("Quiz - PyQt6 + SQLite")
53         title_label.setStyleSheet("font-size: 20px; font-weight: bold; color: black;")
54         header_layout.addWidget(title_label)
55         header_layout.addStretch()
56         self.main_layout.addLayout(header_layout)
57         # -----

```

```

56         self.question_label = QLabel()
57         self.question_label.setStyleSheet("font-size: 18px; font-weight: bold; color: black;")
58         self.main_layout.addWidget(self.question_label)
59
60         self.options_group = QButtonGroup()
61         self.options_layout = QVBoxLayout()
62         self.main_layout.addLayout(self.options_layout)
63
64         self.submit_btn = QPushButton("Submit")
65         self.submit_btn.setStyleSheet(self.button_style("#9d9b9f"))
66         self.submit_btn.clicked.connect(self.check_answer)
67         self.main_layout.addWidget(self.submit_btn)
68
69         self.load_question()
70
71     def button_style(self, color):
72         return f"""
73         QPushButton {{
74             background-color: {color};
75             color: black;
76             font-weight: bold;
77             padding: 10px 16px;
78             border: none;
79             border-radius: 8px;
80         }}
81
82         QPushButton:hover {{
83             background-color: #dfefff;
84             color: black;
85         }}
86         """
87
88     def load_question(self):
89         for btn in self.options_group.buttons():
90             self.options_group.removeButton(btn)
91             btn.deleteLater()
92
93         q_data = self.questions_data[self.current_index]
94         question_text, option_a, option_b, option_c, option_d, _ = q_data
95
96         self.question_label.setText(f"Q{self.current_index + 1}: {question_text}")
97
98         for option in [option_a, option_b, option_c, option_d]:
99             radio_btn = QRadioButton(option)
100             radio_btn.setStyleSheet("font-size: 14px; color: black;")
101             self.options_group.addButton(radio_btn)
102             self.options_layout.addWidget(radio_btn)
103
104     def check_answer(self):
105         selected_btn = self.options_group.checkedButton()
106         if not selected_btn:
107             QMessageBox.warning(self, "No Selection", "Please select an answer.")
108             return
109
110         selected_answer = selected_btn.text()
111         correct_answer = self.questions_data[self.current_index][5]
112
113         if selected_answer == correct_answer:
114             QMessageBox.information(self, "Correct!", "That's the right answer!")
115             self.current_index += 1
116             if self.current_index < len(self.questions_data):
117                 self.load_question()
118             else:
119                 QMessageBox.information(self, "Quiz Finished", "You have completed the quiz!")
120                 self.close()
121         else:
122             QMessageBox.critical(self, "Wrong!", f"The correct answer was: {correct_answer}\nPlease try again.")
123
124     def goto_main(self):
125         if self.main_window:
126             self.main_window.show()
127         self.close()

```



សូមអរគុណ

