



សាកលវិទ្យាល័យ ន័រតុន

NORTON UNIVERSITY

ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

Subject: Introduction Artificial Intelligence (AI)

Taught by Lecturer: Sek Socheat

Assignment

Group Member :

1. ប៉េង ម៉ាលាក់ ID: B20235253
2. ឬ ស៊ីណាន់ ID: B20242856
3. អ៊ិន សុខហេង ID: B20236655
4. សាំង មេសា ID: B20243361



មាតិកា

១. ចំណងបើកកម្មវិធី APPLICATION TITLE
២. ក្រុមនិស្សិត TEAM MEMBERS
៣. គោលបំណងនៃកម្មវិធី PROJECT OBJECTIVE
៤. បច្ចេកវិទ្យាដែលប្រើ TECHNOLOGIES USED
៥. មុខងារសំខាន់ៗនៃកម្មវិធី MAIN FEATURES
៦. មុខងារ AI ដែលបានបញ្ចូល AI FUNCTIONALITY
៧. លក្ខណៈពិសេសនៃកម្មវិធី Special Features
៨. រូបភាព UI គំរូ Optional Mockup
៩. ផែនការអភិវឌ្ឍ Development Timeline
១០. ទិន្នន័យគាំបត់ Supporting Materials



១. ចំណងបើកម្មវិធី APPLICATION TITLE

NoteQ ជាកម្មវិធីអភិវឌ្ឍដោយបច្ចេកវិទ្យា AI សម្រាប់កុំព្យូទ័រ ដែលត្រូវបានរចនាឡើងដើម្បីជួយសិស្ស និងអ្នកសិក្សាឯករាជ្យក្នុងការបង្រៀនមាតិកាបង្រៀនរបស់ពួកគេឲ្យក្លាយជាឧបករណ៍សិក្សាដែលមានប្រសិទ្ធភាព។ មិនថាអ្នកវាយដោយដៃ អាប៊ីឡូតឯកសារ ឬនិយាយមាត់ក៏ដោយ NoteQ នឹងដំណើរការបញ្ចេញនូវសំណួរដើម្បីចោទសួរ និងផ្តល់នូវលំហាត់ខ្លីៗ ដើម្បីជួយឲ្យយល់បានងាយ សូត្រចាំបានល្អ និងអនុវត្តបានប្រសើរជាងមុន។

២. ក្រុមមេសាមីស្រី TEAM MEMBERS

ល.រ	ឈ្មោះ	លេខសម្គាល់	តួនាទី
១	ប៉េង ម៉ាលាក់	B20235253	Full Stack developer
២	ប្តូ ស៊ីណាន់	B20242856	Ux/UI Designer



៣	អ៊ាន សុខហេង	B2023665	QA
៤	សាំង មេសា	B20243361	Data Engineer

៣. គោលបំណងនៃកម្មវិធី (Project Objective)

គម្រោងនេះមានបំណងក្នុងការបង្កើតកម្មវិធីមួយប្រើប្រាស់ភាសា Python និង PyQt6 ដែលអាចឱ្យអ្នកប្រើបញ្ចូលចំណងជើងមេរៀន ឬឯកសារដែលផ្ទុកជាfile (PDF, Word, ឬ Text) វានឹងបង្កើតសំណួរ និងចម្លើយដោយស្វ័យប្រវត្តិ ដោយផ្អែកលើខ្លឹមសារមេរៀនដែលបានផ្តល់ ដោយប្រើបច្ចេកវិទ្យា AI។ គោលដៅនៃកម្មវិធីគឺដើម្បីជួយសិស្សអាចរៀន ឬរៀបចំសំណួរត្រៀមប្រឆាំងបានងាយស្រួល និងរហ័ស។ កម្មវិធីនេះនឹងជួយពង្រឹងការសិក្សានិងការអនុវត្តន៍បានល្អប្រសើរជាងមុនតាមរយៈការបង្កើតសំណួរដែលពាក់ព័ន្ធទៅនឹងមេរៀន។

៤. បច្ចេកវិទ្យាដែលប្រើ (Technologies Used)

. Backend Language	Python 3.10+
. 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

៥. មុខងារសំខាន់ៗនៃកម្មវិធី (Main Features)

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
៦	បកប្រែភាសា	បកប្រែសំណួរនិងចម្លើយទៅជាភាសាផ្សេងៗប្រើ API បកប្រែ
៧	លំហាត់តេស្ត	ធ្វើតេស្តសំណួរដែលបានបង្កើតនិងទទួលបានពិន្ទុដោយស្វ័យប្រវត្តិ

៦. មុខងារ AI ដែលបានបញ្ចូល (AI Functionality):

- ◆ បង្កើតសំណួរពីអត្ថបទ (Text-to-Question Generation):

AI នឹងប្រើ NLP (Natural Language Processing) ដើម្បីយល់អត្ថន័យនៃអត្ថបទឬឯកសារ ហើយបង្កើតសំណួរដែលទាក់ទងនឹងមាតិកានោះ។

- ◆ បង្កើតចម្លើយ (Answer Generation):

ប្រសិនបើអ្នកប្រើបានចម្លើយផងដែរ AI នឹងបញ្ចេញចម្លើយត្រឹមត្រូវសម្រាប់សំណួរដែលបានបង្កើត។

៧. 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។



៨. រូបភាព UI គំរូ (Optional Mockup)

- អ្នកអាចប្រើ Canva, Figma ឬ Qt Designer ដើម្បីបង្កើត Mockup
- បង្កើតផ្ទាំង Home, Input, Output, Settings, Save

៩. ផែនការការអភិវឌ្ឍ Development Timeline

អាទិត្យ	ការងារ	អ្នកទទួលខុសត្រូវ
១	រៀបចំគម្រោង, រចនាUI	ប៉េង ម៉ាលាក់
២	បង្កើតមុខងារGenerate Lesson + Questions	ឬ ស៊ីណាន់
៣	បញ្ចូលមុខងារ Python Code Checker & រក្សាទុកទិន្នន័យ	អ៊ាន សុខហេង
៤	កែ UI, បញ្ចប់ Manual & សាកល្បង	សាំង មេសា



១០. ទិន្នន័យចាំបាច់ (Supporting Materials)

- OpenAI API Key

<https://platform.openai.com/docs>

- Python Libraries :

- UI Assets: Icons និង Logo

<https://github.com/topics/ui-assets>

I. លំដាប់លំដោយលម្អិតនៃការប្រើប្រាស់កម្មវិធី

1. ទំព័រចូលប្រើ (Login/Register)

- អត្តសញ្ញាណអ្នកប្រើប្រាស់

- បញ្ចូល ឈ្មោះអ្នកប្រើ (Username) និង ពាក្យសម្ងាត់ (Password) ។

- ប៊ូតុង Log in ដើម្បីចូលទៅកាន់ប្រព័ន្ធ។

- ប្រសិនបើអ្នកថ្មី ចុច Register ដើម្បីបង្កើតគណនីថ្មី។



Login

Hi, Welcome! 🤖

Username:

Enter username

Password:

Enter password

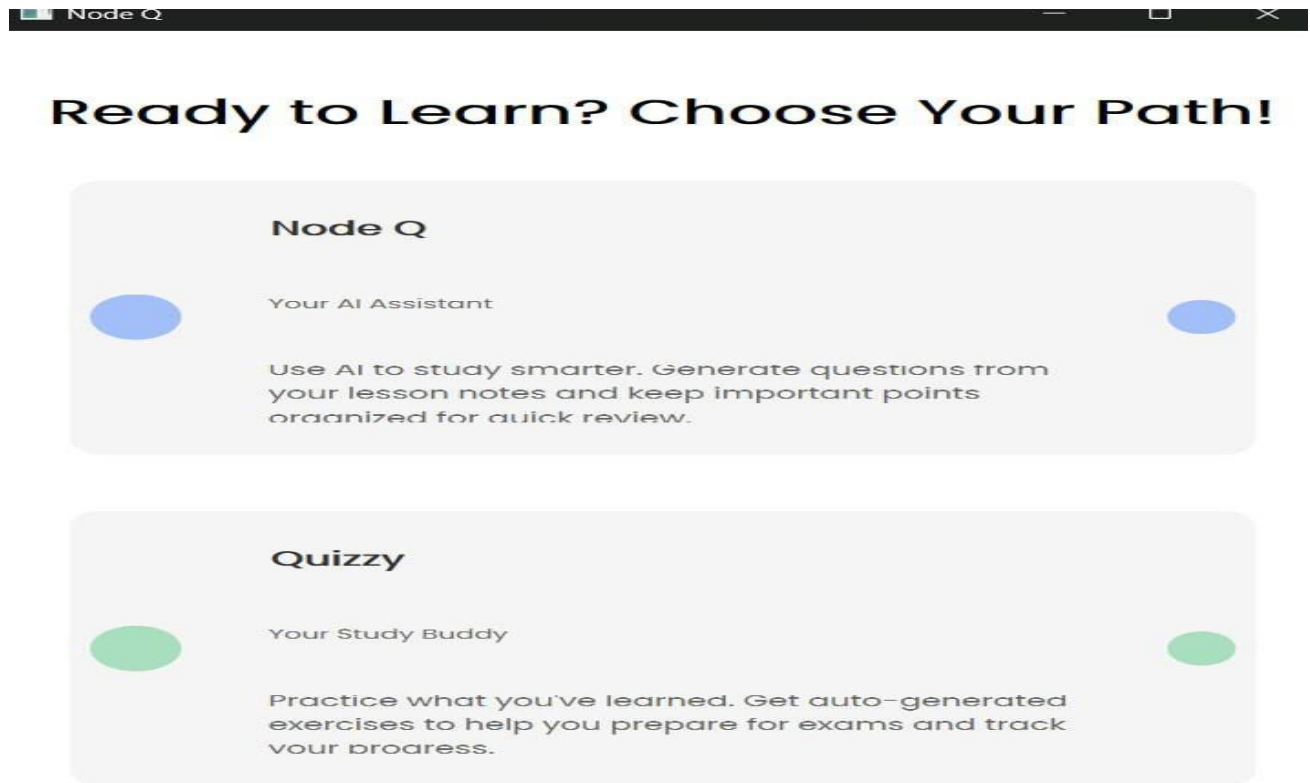
Log in

Register

2. ទំព័រមុខងារសំខាន់ (Dashboard)

បន្ទាប់ពីចូលប្រើជោគជ័យ អ្នកនឹងឃើញអេក្រង់ដែលអនុញ្ញាតឱ្យអ្នកជ្រើសរើសមុខងារដូចខាងក្រោម៖

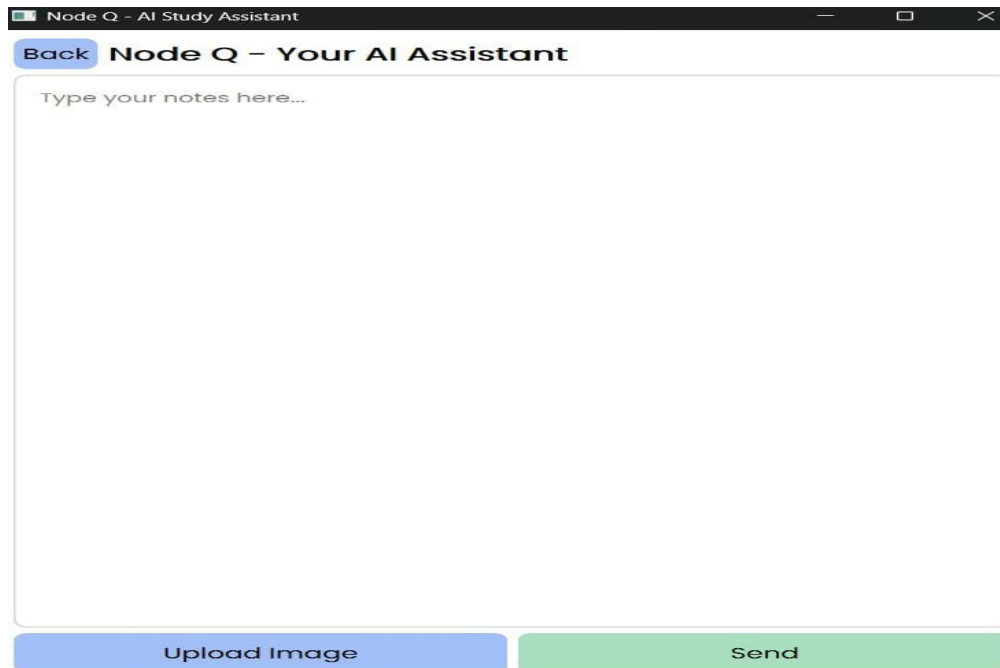
- Node Q (ជំនួយការសិក្សាដោយ AI)
 - ប្រើប្រាស់ AI ដើម្បីបង្កើតសំណួរពីកំណត់ត្រាមេរៀនរបស់អ្នក។
 - រៀបចំចំណុចសំខាន់ៗសម្រាប់មើលឡើងវិញយ៉ាងឆាប់រហ័ស។
- Quizzy (ដៃគូធ្វើលំហាត់)
 - ទទួលបានលំហាត់ដែលបង្កើតដោយស្វ័យប្រវត្តិ ដើម្បីជួយរៀបចំការប្រឡង។
 - តាមដានវឌ្ឍនភាពរាល់ពេលដែលអ្នកធ្វើលំហាត់។



3. ការប្រើប្រាស់ Node Q (AI Study Assistant)

ប្រសិនបើអ្នកជ្រើសរើស Node Q អ្នកនឹងឃើញអន្តរមុខដូចខាងក្រោម៖

- ប៊ូតុង Back ដើម្បីត្រឡប់ទៅទំព័រមុន។
- ប្រអប់បញ្ចូលអត្ថបទ ដែលអ្នកអាចវាយបញ្ចូលកំណត់ត្រាមេរៀនរបស់អ្នក។
- ប៊ូតុង Upload Image ដើម្បីផ្ញើកំណត់ត្រាភាព (ឧទាហរណ៍៖ សន្លឹកកំណត់ត្រាដែលសរសេរដៃ)។
- ប៊ូតុង Send ដើម្បីដាក់ស្នើឯកសារទៅ AI ដើម្បីបង្កើតសំណួរ ឬសង្ខេប។



4. ទំព័រប្រឡង (Quiz Interface)

ប្រសិនបើអ្នកជ្រើសរើស Quizzy ឬ Node Q បានបង្កើតសំណួរ អ្នកនឹងឃើញអន្តរមុខប្រឡងដូចខាងក្រោម៖

- ចំណងជើងប្រឡង (ឧទាហរណ៍៖ *Quick Quiz - PyQt6 + SQLite*)។
- សំណួរពហុជម្រើស (ឧទាហរណ៍៖ *Q1: What is the capital of France ? *)។
- ជម្រើស៖ Berlin, Madrid, Paris, Rome (ចម្លើយត្រឹមត្រូវគឺ Paris)។
- ប៊ូតុង Submit ដើម្បីដាក់ស្នើចម្លើយ និងទទួលបានលទ្ធផល។



python

Back

Quiz – PyQt6 + SQLite

Q1: What is the capital of France?

☐ Berlin

☐ Madrid

☐ Paris

☐ Rome

Submit

លំដាប់ដំណើរការពេញលេញ ៖

1. ចូលប្រើប្រព័ន្ធ (Login/Register)
2. ជ្រើសរើសមុខងារ (Node Q ឬ Quizzy)
3. បញ្ចូលមាតិកា (អត្ថបទ ឬរូបភាព) ហើយចុច Send បើប្រើ Node Q
4. ទទួលសំណួរ/លំហាត់ ពី AI ហើយឆ្លើយតប
5. ដាក់ស្នើចម្លើយ ដើម្បីឃើញលទ្ធផល

Souces Code



សំណើអភិវឌ្ឍន៍កម្មវិធី និស្សិតឆ្នាំទី២ ឆមាសទី២ (២០២៥)

```

1  # Import necessary Python modules for UI components and functionality
2  from PyQt5.QtWidgets import QWidget, QLabel, QLineEdit, QPushButton, QVBoxLayout, QMessageBox, QCheckBox
3  from PyQt5.QtDatabase 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 PyQt5.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()
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("<span style='font-size: 16px; color: #000;'>Username:</span>")
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("<span style='font-size: 16px; color: #000;'>Password:</span>")
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.goto_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 goto_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()

```



សំណើអភិវឌ្ឍន៍កម្មវិធី និស្សិតឆ្នាំទី២ ឆមាសទី២ (២០២៥)

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



សំណើអភិវឌ្ឍន៍កម្មវិធី និស្សិតឆ្នាំទី២ ឆមាសទី២ (២០២៥)

```
1 from PyQt5.QtWidgets import (
2     QMainWindow, QLabel, QWidget, QVBoxLayout, QHBoxLayout,
3     QGraphicsDropShadowEffect, QPushButton
4 )
5 from PyQt5.QtCore import Qt
6 from PyQt5.QtGui import QPixmap, QColor
7 from ui.quizzy import QuizWindow
8 from ui.notes_al import NoteQWindow # Import the NoteQWindow 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.setVOffset(5)
32         shadow.setColor(QColor("#aaaaaa"))
33         card_container.setGraphicsEffect(shadow)
34
35         layout = QVBoxLayout()
36         layout.setSpacing(10)
37         layout.setContentsMargins(0, 0, 0, 0)
38
39         icon_container = QLabel()
40         icon_container.setFixedSize(40, 40)
41         icon_container.setStyleSheet(f"""
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(f"""
77             QPushButton {{
78                 background-color: {button_color};
79                 border: none;
80                 border-radius: 15px;
81             }}
82         """)
83         layout.addWidget(arrow_button)
84
85         card_container_layout = QHBoxLayout(card_container)
86         card_container_layout.addLayout(layout)
87
88         main_layout = QVBoxLayout(self)
89         main_layout.addWidget(card_container)
90
91 class MainWindow(QMainWindow):
92     def __init__(self):
93         super().__init__()
94         self.setStyleSheet("""
95             QMainWindow {
96                 background-color: white;
97             }
98         """)
99         self.setWindowTitle("Node Q")
100         self.resize(550, 600)
101
102         central_widget = QWidget()
103         main_layout = QVBoxLayout(central_widget)
104         main_layout.setAlignment(Qt.AlignmentFlag.AlignTop)
105         main_layout.setSpacing(30)
106         main_layout.setContentsMargins(20, 50, 20, 20)
107
108         title_label = QLabel("Ready to Learn? Choose Your Path!")
109         title_label.setStyleSheet("font-size: 30px; font-weight: 700; color: black;")
110         main_layout.addWidget(title_label)
111
112         node_q_card = CardWidget(
113             "image/material_symbols_quiz-rounded.png",
114             "Node Q",
115             "Your AI Assistant",
116             "Use AI to study smarter. Generate questions from your lesson notes and keep important points organized for quick review.",
117             "#a1887f" # light purple
118         )
119         node_q_card.mousePressEvent = lambda event: self.open_node_q_window()
120         main_layout.addWidget(node_q_card)
121
122         quizzy_card = CardWidget(
123             "image/material_symbols_quiz-rounded.png",
124             "Quizzy",
125             "Project Manager",
126             "Practice what you've learned. Get auto-generated exercises to help you prepare for exams and track your progress.",
127             "#a1887f" # light green
128         )
129         quizzy_card.mousePressEvent = lambda event: self.open_quiz()
130         main_layout.addWidget(quizzy_card)
131
132         self.setCentralWidget(central_widget)
133
134     def open_node_q_window(self):
135         self.node_q_window = NoteQWindow(main_window=self)
136         self.node_q_window.show()
137
138     def open_quiz(self):
139         self.quiz_window = QuizWindow(main_window=self)
140         self.quiz_window.show()
141         self.hide()
```




សំណើអភិវឌ្ឍន៍កម្មវិធី និស្សិតឆ្នាំទី២ ឆមាសទី២ (២០២៥)

```

1 from PyQt5.QtWidgets import (
2     QMainWindow, QWidget, QVBoxLayout, QHBoxLayout,
3     QLabel, QPushButton, QRadioButton, QMessageBox, QButtonGroup
4 )
5 from PyQt5.QtCore import Qt
6
7 class QuizWindow(QMainWindow):
8     def __init__(self, main_window=None):
9         super().__init__()
10        self.main_window = main_window  # Reference to MainWindow
11
12        self.resize(500, 400)
13        self.setStyleSheet("background-color: white;")
14
15        # Connect to DB and get questions
16        from db_database import Database  # Import here to avoid circular issues
17        self.db = Database()
18        self.questions_data = self.db.get_all_questions()
19        self.current_index = 0
20
21        if not self.questions_data:
22            QMessageBox.critical(self, "Error", "No questions found in the database.")
23            self.close()
24            return
25
26        central_widget = QWidget()
27        self.setCentralWidget(central_widget)
28        self.main_layout = QVBoxLayout(central_widget)
29        self.main_layout.setContentsMargins(20, 20, 20, 20)
30        self.main_layout.setSpacing(15)
31
32        # Back button and header layout
33        header_layout = QHBoxLayout()
34        back_btn = QPushButton("Back")
35        back_btn.setStyleSheet(
36            """
37            QPushButton {
38                background-color: #a3bffa;
39                color: black;
40                padding: 5px;
41                border-radius: 8px;
42            }
43            QPushButton:hover {
44                background-color: #dfefff;
45            }
46            """
47        )
48        back_btn.clicked.connect(self.goto_main)
49        header_layout.addWidget(back_btn)
50
51        title_label = QLabel("Quiz - PyQt5 + SQLite")
52        title_label.setStyleSheet("font-size: 20px; font-weight: bold; color: black;")
53        header_layout.addWidget(title_label)
54        self.main_layout.addLayout(header_layout)
55
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("background-color: #a9d0d9; color: black;")
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 15px;
78                border: none;
79                border-radius: 8px;
80            }}
81            QPushButton:hover {{
82                background-color: #dfefff;
83                color: black;
84            }}
85            """
86
87        def load_question(self):
88            for btn in self.options_group.buttons():
89                self.options_group.removeButton(btn)
90                btn.deleteLater()
91
92            q_data = self.questions_data[self.current_index]
93            question_text, option_a, option_b, option_c, option_d, _ = q_data
94
95            self.question_label.setText(f"Q{self.current_index + 1}: {question_text}")
96
97            for option in [option_a, option_b, option_c, option_d]:
98                radio_btn = QRadioButton(option)
99                radio_btn.setStyleSheet("font-size: 14px; color: black;")
100                self.options_group.addButton(radio_btn)
101                self.options_layout.addWidget(radio_btn)
102
103        def check_answer(self):
104            selected_btn = self.options_group.checkedButton()
105            if not selected_btn:
106                QMessageBox.warning(self, "No Selection", "Please select an answer.")
107                return
108
109            selected_answer = selected_btn.text()
110            correct_answer = self.questions_data[self.current_index][5]
111
112            if selected_answer == correct_answer:
113                QMessageBox.information(self, "Correct!", "That's the right answer!")
114                self.current_index += 1
115                if self.current_index < len(self.questions_data):
116                    self.load_question()
117                else:
118                    QMessageBox.information(self, "Quiz Finished", "You have completed the quiz!")
119                    self.close()
120            else:
121                QMessageBox.critical(self, "Wrong!", f"The correct answer was: {correct_answer}. Please try again.")
122
123        def goto_main(self):
124            if self.main_window:
125                self.main_window.show()
126                self.close()
127
128

```




II. លទ្ធផល (Results)

ក្រោយពេលអភិវឌ្ឍ និងសាកល្បងកម្មវិធី NoteQ អ្នកប្រើអាចទទួលបានលទ្ធផលដូចខាងក្រោម៖

ការនាំចូលឯកសារ (File Import)

អ្នកប្រើអាចជ្រើសឯកសារ PDF, Word (.docx) ឬ Text (.txt) ដើម្បីទាញយកមាតិកា។

កម្មវិធីអានឯកសារនិងបម្លែងមាតិកាទៅអត្ថបទសុទ្ធសាធ។

ការបញ្ចូលចំណងជើងមេរៀន (Lesson Title Input)

ប្រសិនបើអ្នកមិនមានឯកសារ អ្នកអាចវាយចំណងជើងមេរៀន ដូចជា “Computer Networks – OSI

Model” ហើយ NoteQ នឹងប្រើ AI ដើម្បីស្វែងរក និងបង្កើតសំណួរ-ចម្លើយ។

ការបង្កើតសំណួរ និងចម្លើយដោយស្វ័យប្រវត្តិ (Auto Q&A Generation)

AI នឹងវិភាគមាតិកា → កំណត់ប្រធានបទសំខាន់ៗ → បង្កើតសំណួរ និងចម្លើយ។

សំណួរអាចមានទម្រង់ Multiple Choice, True/False, Short Answer។

ការផ្ទុកលទ្ធផល (Save Output)

អ្នកអាចរក្សាទុកសំណួរ និងចម្លើយជា Text File, Word File ឬចម្លងទៅ Clipboard ដើម្បីប្រើបន្ត។

III. ការពិភាក្សា (Discussions)

ការពិភាក្សាប្រសើរនៅលើ NoteQ អាចបែងចែកបានជា៖

_ អត្ថប្រយោជន៍ (Advantages)

_ សន្សំពេលវេលា – មិនចាំបាច់អានឯកសារទាំងមូល ហើយសរសេរសំណួរដោយដៃ។



សំណើអភិវឌ្ឍន៍កម្មវិធី និងស្ថិតភ្នំទី២ ឆមាសទី២ (២០២៥)

- _ ភាពងាយស្រួល – UI ត្រូវបានរចនាឲ្យមានប៊ូតុង និងម៉ឺនុយច្បាស់។
- _ ភាពបត់បែន – អាចប្រើបានទាំងសិស្ស, គ្រូ, និងអ្នករៀនដោយខ្លួនឯង។
- _ កម្រិតកំណត់ (Limitations)

AI អាស្រ័យលើគុណភាពមាតិកា – ប្រសិនបើមាតិកាមានការប្រែប្រួលវិភាសា ឬគ្មានសេចក្តីរៀបចំល្អ សំណួរអាចមិនត្រឹមត្រូវ។

មិនគាំទ្រឯកសារដែលមានរូបភាពតែប៉ុណ្ណោះ (Image-based PDFs) ។

តម្រូវឲ្យមានអ៊ីនធឺណិតសម្រាប់ប្រើ AI។

ឱកាសអភិវឌ្ឍន៍បន្ថែម (Future Improvements)

បន្ថែម Flashcard Mode សម្រាប់រៀនឡើងវិញលើកក្រោយ។

បន្ថែម Quiz Test Mode សម្រាប់សាកល្បងចម្លើយភ្លាមៗ។

កែលម្អការគាំទ្រឯកសារ Image-based PDF ដោយ OCR។

IV. សន្និដ្ឋាន (Conclusions)

- _ កម្មវិធី NoteQ បង្ហាញពីសក្តានុពលនៃការប្រើ Python + PyQt6 + AI ក្នុងការកែលម្អវិធីសិក្សា។
- _ វាអាចបម្លែងមាតិកាអប់រំទៅជាសំណួរ និងចម្លើយដោយស្វ័យប្រវត្តិ។
- _ ជួយសន្សំពេល និងធ្វើឲ្យការរៀនមានប្រសិទ្ធភាព។
- _ មានឱកាសអភិវឌ្ឍន៍បន្ថែមដើម្បីធ្វើឲ្យមានភាពពេញលេញ និងប្រើបានទូលំទូលាយ។

ឯកសារយោង (References)



1. Python Software Foundation. Python Language Reference. <https://www.python.org/>
2. Riverbank Computing. PyQt6 Documentation.
<https://www.riverbankcomputing.com/static/Docs/PyQt6/>
3. Brownlee, J. (2019). Deep Learning for Natural Language Processing. Machine Learning Mastery
4. Vaswani, A., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems.
5. Scikit-learn developers. Machine Learning in Python. . <https://scikit-learn.org/>