# Import necessary libraries

from PyQt5.QtWidgets import \*

from PyQt5.QtCore import \*

from PyQt5.QtWebEngineWidgets import \*

import sys

# Create a main window class

class MainWindow(QMainWindow):

# Constructor of this class

def \_init\_(self):

super(MainWindow, self).\_init\_()

# To provide a widget for viewing and editing web documents:

self.browser = QWebEngineView()

# To set default browser homepage as google homepage:

self.browser.setUrl(QUrl("http://www.google.com"))

# To set browser as central widget of main window:

self.setCentralWidget(self.browser)

# To open browser in a maximized window:

self.showMaximized()

# To create a navigation bar:

navbar = QToolBar()

navbar.adjustSize()

# To add the navigation bar to the browser:

self.addToolBar(navbar)

# To add back button within navigation bar:

back\_btn = QAction('⮜', self)

back\_btn.triggered.connect(self.browser.back)

navbar.addAction(back\_btn)

# To add forward button within navigation bar:

forward\_btn = QAction('⮞', self)

forward\_btn.triggered.connect(self.browser.forward)

navbar.addAction(forward\_btn)

# To add reload button within navigation bar:

reload\_btn = QAction('⟳', self)

reload\_btn.triggered.connect(self.browser.reload)

navbar.addAction(reload\_btn)

# To add URL bar within navigation bar:

self.url\_bar = QLineEdit()

self.url\_bar.returnPressed.connect(self.open\_url)

navbar.addWidget(self.url\_bar)

self.browser.urlChanged.connect(self.update\_url)

# To navigate to desired URL specified within URL bar:

def open\_url(self):

url = self.url\_bar.text()

self.browser.setUrl(QUrl(url))

# To update the URL bar contents when navigated from one page to another:

def update\_url(self, q):

self.url\_bar.setText(q.toString())

# To call constructor of the C++ class QApplication:

# Here, sys.argv is used to initialize the QT application

app = QApplication(sys.argv)

# To specify name of the browser:

QApplication.setApplicationName("My\_Explorer")

# To create an object of MainWindow class defined above:

window = MainWindow()

# To run the main event loop and wait until exit() is called:

app.exec()