**Code For File Exploler Application:**

Creating a file explorer application in C++ involves using a graphical user interface (GUI) library to handle window creation, file browsing, and user interactions. One popular choice for such tasks is the Qt framework. Below is a simple example of a file explorer application using Qt.

**Prerequisites:**

1. Install Qt: You can download and install Qt from here.
2. Set up your development environment to compile and run Qt applications.

**Example Code:**

1. **Create a Qt Project:** Use Qt Creator to create a new Qt Widgets Application.
2. **Edit the main.cpp file:**

cpp

Copy code

#include <QApplication>

#include "mainwindow.h"

int main(int argc, char \*argv[])

{

QApplication app(argc, argv);

MainWindow window;

window.show();

return app.exec();

}

1. **Create the main window header file (mainwindow.h):**

cpp

Copy code

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QFileSystemModel>

#include <QTreeView>

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private:

QFileSystemModel \*model;

QTreeView \*treeView;

};

#endif // MAINWINDOW\_H

1. **Create the main window implementation file (mainwindow.cpp):**

cpp

Copy code

#include "mainwindow.h"

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent), model(new QFileSystemModel(this)), treeView(new QTreeView(this))

{

model->setRootPath(QDir::rootPath());

treeView->setModel(model);

treeView->setRootIndex(model->index(QDir::rootPath()));

setCentralWidget(treeView);

resize(800, 600);

}

MainWindow::~MainWindow()

{

delete model;

delete treeView;

}

1. **Edit the project file (projectname.pro):**

Make sure your .pro file includes the necessary Qt modules.

pro

Copy code

QT += core gui

greaterThan(QT\_MAJOR\_VERSION, 4): QT += widgets

TARGET = FileExplorer

TEMPLATE = app

SOURCES += main.cpp \

mainwindow.cpp

HEADERS += mainwindow.h

**Explanation:**

* **main.cpp**: Initializes the Qt application and shows the main window.
* **mainwindow.h**: Declares the main window class, which includes a file system model and a tree view.
* **mainwindow.cpp**: Implements the main window, setting up the file system model and tree view to display the file system. The root path is set to the root directory of the file system.
* **projectname.pro**: Specifies the project configuration, including the Qt modules to use.

**Building and Running the Application:**

1. Open your project in Qt Creator.
2. Configure the project for your preferred build environment.
3. Build and run the application.

This will open a window displaying a tree view of the file system, allowing you to navigate through directories and files. You can expand upon this basic example to add more features like file operations (copy, move, delete), search functionality, and more.