# PATTERN CODERS

## Setup Instructions

### 1. Install Necessary Dependencies:

* Install [Qt](https://www.qt.io/download-qt-installer-oss)
* Install [CMake](https://cmake.org/download/)
* Install [MinGW](https://sourceforge.net/projects/mingw/) or another compiler

### 2. Clone the Project:

git clone <https://github.com/codingburgas/10grade-christmas-luck-pattern-coders.git>

Open the project in an IDE (preferably QtCreator).

### 3. Build and Run the Application:

Build, run the application, and expand your knowledge!

## Communication Between Front-End and Back-End

### Exposing an Object to QML for Communication:

/\* src/application.cpp : 105 \*/  
engine.rootContext()->setContextProperty("application", this);

### UI Update Using Qt Signal System:

Example:

/\* src/application.cpp : 132 \*/  
emit message(QString::fromStdString(m.title), QString::fromStdString(m.description), QString::fromStdString(m.type));

"Connections" object listens to signals on the application object and invokes the onMessage function.

/\* assets/ui/Application.qml : 89 \*/  
Connections{  
 target: application  
  
 function onMessage(title, description, type){  
 let messageComponent = Qt.createComponent( Qt.resolvedUrl("Message.qml") )  
  
 let waitFunction = () => {  
 if (messageComponent.status == Component.Ready){  
 let message = messageComponent.createObject(  
 mainWindow,  
 {  
 title: title,  
 description: description,  
 type: type  
 });  
  
 message.destroy(2000)  
 } else if (newComponent.status === Component.Loading) {  
 newComponent.statusChanged.connect(waitFunction); // Connect to statusChanged if still loading  
 }  
 }  
  
 waitFunction();  
 }  
}

### Handling User Requests:

/\* assets/ui/MainPage.qml : 627 \*/  
application.searchWords(searchInput.text, property2ComboBox.getProperty(), caseSensitiveCheckBox.checked, (propertyHasToComboBox.currentIndex==1), (propertyHasToComboBox.currentIndex==2))

In this example, the searchWords method of the application object is called to search words with specific properties and update the UI.

## Project Features

### 1. User-Friendly Interface:

The application provides a user-friendly interface designed with QML, ensuring an intuitive and seamless user experience. The interface is responsive and adapts to different screen sizes, enhancing accessibility.

### 3. Advanced Search Functionality:

Users can search for specific data using various filters and criteria. The search functionality is powered by the searchWords method, which efficiently processes user queries and displays relevant results.

## Our Team Members

* [Petar Kostov](https://github.com/PTKostov22)
* [Vinichenko Oleksandr Oleksandrov](https://github.com/VOOleksandrov22)
* [Denis Pirig](https://github.com/DIPirig22)