

I have organized the project into two main parts:

1. ***OpenOCRCorrect (with Bold/Unbold, Superscript, Subscript, and Improved UI):***

* This will be the core of the OCR correction tool with enhanced functionality.

2. ***Text Editor (Qt-based):***

* This will be the basic text editor with bold/unbold, superscript, subscript, and an "About Me" feature.

GitHub Repository:

I'll create a new GitHub repository to house both parts of the project, organized into appropriate folders:

* [OpenOCRCorrect/](#): Contains the improved OpenOCRCorrect code.

* [TextEditor/](#): Contains the Qt-based text editor code.

1. OpenOCRCorrect Enhancements

*** *Clone the Repository:***

```
bash
```

```
git clone https://github.com/rohitsuja22/OpenOCRCorrect.git
```

```
cd OpenOCRCorrect
```

*** *Install Dependencies:***

```
bash
```

```
pip install -r requirements.txt
```

*** *Add Bold/Unbold Functionality:***

1. ****Modify ocrorrect.py:****

* Add the following imports at the top:

```
python
```

```
from rich.panel import Panel
```

```
from rich.text import Text
```

```
from rich.console import Console
```

* Create a function to toggle bold/unbold:

```
python

def toggle_bold(text):

    """Toggles bold formatting in the given text."""

    console = Console()

    if console.is_jupyter:

        return f"***{text}***"

    return f"[{text}]"
```

* Update the process_text function (or where you handle the output) to use the toggle_bold function. For example:

```
python

def process_text(text, lang, model):

    # ... (rest of the function)

    if 'bold' in corrections:

        text = toggle_bold(text)

    return text
```

*** *Add Superscript/Subscript Functionality:***

1. ****Create superscript_subscript.py:****

```
python

def superscript(text):

    """Applies superscript formatting to the given text."""

    return f"<sup>{text}</sup>"

def subscript(text):
```

```
"""Applies subscript formatting to the given text."""
```

```
return f"<sub>{text}</sub>"
```

2. ****Update ocrorrect.py:****

- * Import superscript and subscript from superscript_subscript.py.

- * Modify the process_text function to use these functions based on the detected corrections.

*** *Add a "How to Use" Section to the README.md:***

- * Explain how to run the script, including the new bold/unbold and superscript/subscript features.

2. Qt Text Editor Development

*** *Install Qt Creator:***

- * Download and install Qt Creator from <https://www.qt.io/>

*** *Create a New Project:***

- * Open Qt Creator.

- * Go to "File" -> "New File or Project."

- * Choose "Qt Widgets Application" and click "Choose."

- * Name your project (e.g., "TextEditor").

- * Accept the default kit and UI settings.

*** *Design the UI (mainwindow.ui):***

- * Open the mainwindow.ui file.

- * Add a QTextEdit for the main editing area.

- * Add a QToolBar for the toolbar buttons.

- * Add a QMenuBar for the "About Me" menu.

- * Create actions for Bold, Unbold, Superscript, and Subscript on the toolbar.

* Create an "About Me" action on the menu bar.

*** *Implement the Logic (mainwindow.cpp):***

*** *Header:***

c++

#include "mainwindow.h"

#include "ui_mainwindow.h"

#include <QTextCharFormat>

#include <QFileDialog>

#include <QMessageBox>

*** *Constructor:***

c++

MainWindow::MainWindow(QWidget *parent)

: QMainWindow(parent), ui(new Ui::MainWindow) {

ui->setupUi(this);

// Connect signals to slots (bold, unbold, superscript, subscript, about me)

}

*** *Slots:***

```c++

void MainWindow::onBold() {

// Get current text cursor and format

QTextCursor cursor = ui->textEdit->textCursor();

QTextCharFormat format = cursor.charFormat();

// Toggle bold font

```
format.setFontWeight(format.fontWeight() == QFont::Bold ? QFont::Normal : QFont::Bold);
cursor.mergeCharFormat(format);
ui->textEdit->setTextCursor(cursor);
}
```

```
void MainWindow::onUnbold() {
 // Similar to onBold but sets font weight to Normal
}
```

```
void MainWindow::onSuperscript() {
 // Get current text cursor and format
 QTextCursor cursor = ui->textEdit->textCursor();
 QTextCharFormat format = cursor.charFormat();
 // Toggle superscript
 format.setVerticalAlignment(format.verticalAlignment() == QTextCharFormat::AlignSuperScript ?
 QTextCharFormat::AlignNormal : QTextCharFormat::AlignSuperScript);
 cursor.mergeCharFormat(format);
 ui->textEdit->setTextCursor(cursor);
}
```

```
void MainWindow::onSubscript() {
 // Similar to onSuperscript but sets vertical alignment to AlignSubScript
}
```

```
void MainWindow::showAboutMe() {
```

```
// Create a QMessageBox

QMessageBox msgBox;

msgBox.setWindowTitle("About Me");

// Set the text to display your resume (using HTML or plain text)

msgBox.setText("Your Resume Here (HTML or Text)");

// Set the font and styling of the message box

// ... (use QFont and other styling options)

msgBox.exec();
```