1. Objectives:

- Reform the user interface into six groups: File, Basic, Filtering, Advances for operations, Global, and Regional adjustments.
- Install OpenCV with GPU support, ensuring compatibility with CuDNN and integration with Qt Creator.
- Organize the main.cpp file by separating it into multiple header and cpp files.
- Debug issues arising from the separation of files, particularly those affecting display functionality.

---

2. Activities:

- Reformed the UI into six distinct groups:
  - **File**: Handling file operations.
  - **Basic**: For basic image adjustments and manipulation.
  - **Filtering**: Applying filters to images.
  - **Advances**: Advanced operations.
  - **Global**: Global adjustments applicable to the whole image.
  - **Regional**: Adjustments targeted at specific regions within the image.
- **OpenCV Installation Attempts**:
  - Attempted to install OpenCV with GPU support using **MingW**, but the installation failed due to MingW not supporting CuDNN.
  - Switched to **MSVC** and **CMake** to build OpenCV with GPU support, and the build was successful. However, linking with **Qt Creator** failed due to CuDNN incompatibility.
  - Updated **Qt Creator to version 6.8.0** with **MSVC 17 2020** to ensure proper support for CuDNN and other GPU functionalities.
  - Reconfigured and rebuilt the **opencv-build** using MSVC as the generator and downloaded the **opencv-contrib** package via Git to support GPU functionality.
- **File Separation**:
  - Separated the previously large **main.cpp** file into three header files and three cpp files, including **main.cpp**, to improve code organization and maintainability.
- **Bug Fixes**:
  - After file separation, several bugs were detected:
    1. The **rectangle drawing function** did not display as expected.
    2. **Pixel information** was being shown in the console instead of the intended location in the UI.
    3. The **loadTxtImage()** function failed to load the optimal image.
  - Found that the **updateImageDisplay()** function, which is responsible for updating the image on the display, was missing from many functions after the file separation. The function calls were added back based on the original source code.
- **Installation Progress**:
  - Successfully installed the **opencv-build** via **Visual Studio** but have not yet tested running it in **Qt Creator**.

---

3. Achievements:

- Completed UI reformation into six well-organized groups for easier navigation and better functionality.
- Successfully built OpenCV with GPU support using MSVC and CMake, with plans to test it further in Qt Creator.
- Organized and modularized the code by splitting the main.cpp file into multiple header and cpp files.
- Fixed several bugs related to image display and pixel information by adding back the missing **updateImageDisplay()** function.

---

4. Problems & Solutions:

- **Problem:** OpenCV with GPU installation failed when using MingW due to lack of CuDNN support.
    - **Solution:** Switched to MSVC and CMake for building OpenCV with GPU, which was successful.
- **Problem:** Linking OpenCV with Qt Creator showed errors due to CuDNN incompatibility.
    - **Solution:** Updated Qt Creator and MSVC, reconfigured OpenCV with opencv-contrib to resolve the issue, and plan to further test the integration.
- **Problem:** After separating the main.cpp file, several bugs emerged, including the rectangle drawing function not displaying, pixel info being shown in the console, and the loadTxtImage() function failing to load the best image.
    - **Solution:** Tracked the issue down to missing **updateImageDisplay()** function calls and reinserted the function calls where necessary, resolving the display and pixel info bugs.