Feature: Keyword / String / Comment Highlighting
File: `CodeEditor.html`

STEP 0 — Open the correct file

👉 Open **`CodeEditor.html`**

STEP 1 — ADD CSS (ONLY HERE)

📍 **Where**

Inside `<style>` block in `CodeEditor.html`
👉 **At the very bottom**, just before `</style>`

1.1 Syntax Highlighting Classes (NEW)

```
/* ===== Syntax Highlighting ===== */

.py-keyword {
  color: #ff5252;   /* Python keywords */
}

.sql-keyword {
  color: #40c4ff;   /* SQL keywords */
}

.spark-keyword {
  color: #ff9100;   /* Spark keywords */
}

.string {
  color: #ffd54f;   /* Strings */
}

.comment {
  color: #9e9e9e;   /* Comments */
  font-style: italic;
}
```

✅ Purpose:
Defines **colors only**.
Does not affect layout or logic.

1.2 Editor Overlay Styling (EXTRA but REQUIRED)

```
/* ===== Syntax Highlight Overlay ===== */

#editor {
  position: absolute;
  top: 45px;                  /* below menu bar */
  left: 10%;
  width: 90%;
  height: calc(100% - 45px);
  background: #1F2833;
  color: #b2b2b2;
  font-family: monospace;
  font-size: 18px;
  white-space: pre-wrap;
  pointer-events: none;       /* IMPORTANT */
  z-index: 1;
  padding: 5px;
}

#data {
  position: relative;
  background: transparent;
  color: transparent;         /* hide raw text */
  caret-color: #4CAF50;       /* visible cursor */
  z-index: 2;
}
```

✅ Purpose:
Creates **textarea + highlighted layer overlay**
This is why highlighting works while typing.

2️⃣ HTML Changes (Overlay Element)

📍 Where

Inside `#upperpart`, **above the textarea**

Added element (NEW)

```
<pre id="editor"></pre>
<textarea name="data" id="data"></textarea>
```

✅ Purpose:

- `<pre id="editor">` → shows highlighted code
- `<textarea id="data">` → handles typing & cursor

⚠️ This is **not optional** — required for highlighting.

3️⃣ JavaScript Changes

📍 Where

Inside `<script>` → **inside** `$(document).ready(function(){ ... })`

3.1 Keyword Configuration (NEW)

Placed **immediately after**:

```
$(document).ready(function(){
/* ===== Keyword Configuration ===== */

const pythonKeywords = [
  'def','return','if','elif','else','for','while','break',
  'continue','import','from','as','class','try','except',
  'finally','with','lambda','pass'
];

const sqlKeywords = [
  'select','from','where','join','inner','left','right',
  'group','by','order','insert','update','delete','create',
  'table','values','distinct'
];

const sparkKeywords = [
  'spark','rdd','dataframe','dataset','map','filter',
  'reduce','collect','parallelize'
];
```

✅ Purpose:
Defines language tokens only.

3.2 `highlightCode()` Function (NEW / REPLACED)

Placed **right after keyword arrays**

```
function highlightCode(code) {

  // escape HTML
  code = code
    .replace(/&/g,'&amp;')
    .replace(/</g,'&lt;')
    .replace(/>/g,'&gt;');

  // strings
  code = code.replace(
```

```
    /("[^"]*"|'[^']*')/g,
    '<span class="string">$1</span>'
  );

  // comments (# and --)
  code = code.replace(
    /(#[^\n]*|--[^\n]*)/g,
    '<span class="comment">$1</span>'
  );

  // python keywords
  pythonKeywords.forEach(word => {
    const regex = new RegExp('\\b(' + word + ')\\b','g');
    code = code.replace(regex,'<span class="py-keyword">$1</span>');
  });

  // sql keywords
  sqlKeywords.forEach(word => {
    const regex = new RegExp('\\b(' + word + ')\\b','gi');
    code = code.replace(regex,'<span class="sql-keyword">$1</span>');
  });

  // spark keywords
  sparkKeywords.forEach(word => {
    const regex = new RegExp('\\b(' + word + ')\\b','gi');
    code = code.replace(regex,'<span class="spark-keyword">$1</span>');
  });

  return code;
}
```

✅ Purpose:
Transforms raw code into highlighted HTML.

3.3 Live Editor Wiring (EXTRA but REQUIRED)

Placed **immediately after `highlightCode()`**

```
$('#data').on('input scroll', function () {
  const code = $(this).val();
  $('#editor').html(highlightCode(code));

  // sync scrolling
  $('#editor').scrollTop($(this).scrollTop());
  $('#editor').scrollLeft($(this).scrollLeft());
});
```

✅ Purpose:

- Updates highlighting on typing
- Keeps scrolling in sync

⚠️ This block **did not exist before** — it was added.

3.4 File Load Highlight Fix (EXTRA)

When reading a file from backend:

❌ Old:

```
$("#editor").text(data);
```

✅ New:

```
$("#editor").html(highlightCode(data));
```

📍 Location:
Inside `read` and `readSampleFile` AJAX success blocks.

✅ Purpose:
Ensures highlighting works for loaded files, not only typed text.