

## Information design

The cheapest, fastest, and most reliable components are those that aren't there.

*Gordon Bell*

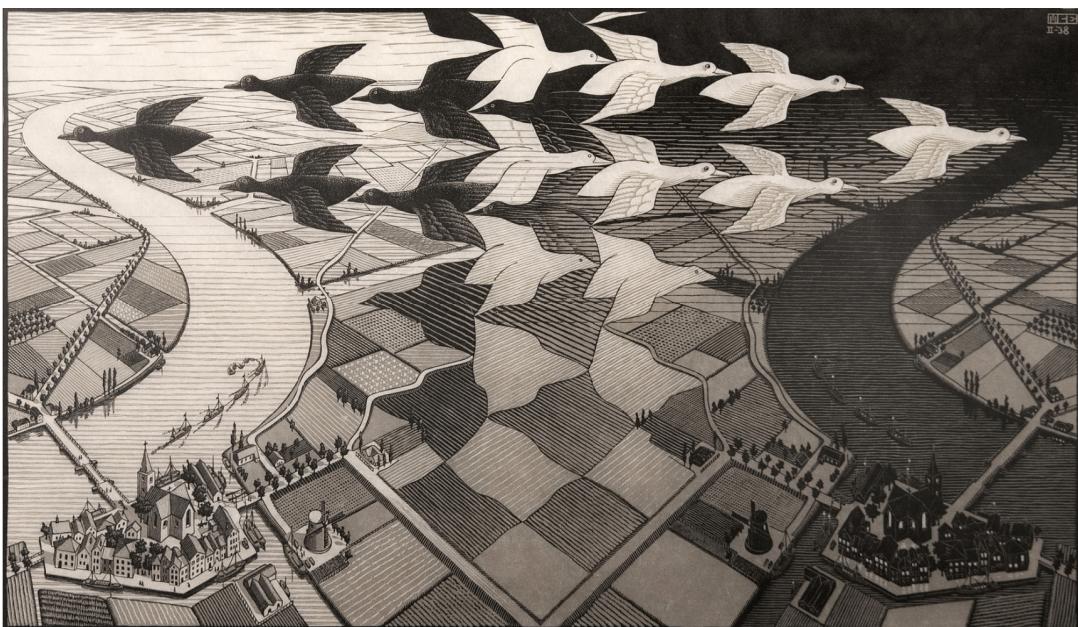


Figure 1: Day and Night, M.C. Escher

---

Score	Criteria
5/10	<b>(Passing - Basic Functionality):</b> Augment your search results with a set of widgets which are activated only when relevant to the current query. For example, searching for 'calculator' should display a widget alongside the regular search results. For the sake of this exercise, widgets are purely visual.
6/10	<b>(Below Average):</b> Implement a caching layer for your search engine. Consider using the Proxy pattern for this.
7/10	<b>(Average):</b> Implement a basic <a href="#">spelling corrector</a> , decoupled from the main query parsing and database interaction logic. For example, searching for 'calclator' should suggest 'calculator' or automatically return results for 'calculator'. Some applicable design patterns here include: Strategy, Decorator, and Facade.

---

5/10 **(Passing - Basic Functionality):** Augment your search results with a set of widgets which are activated only when relevant to the current query. For example, searching for 'calculator' should display a widget alongside the regular search results. For the sake of this exercise, widgets are purely visual.

6/10 **(Below Average):** Implement a caching layer for your search engine. Consider using the Proxy pattern for this.

7/10 **(Average):** Implement a basic [spelling corrector](#), decoupled from the main query parsing and database interaction logic. For example, searching for 'calclator' should suggest 'calculator' or automatically return results for 'calculator'. Some applicable design patterns here include: Strategy, Decorator, and Facade.

---

Continued on next page

---

**Table 1 – continued from previous page**

Score	Criteria
8/10	<p><b>(Good):</b> Use the result set to activate the context-aware widgets. Some examples:</p> <ul style="list-style-type: none"> <li>• Results are mostly .log files → widget showing "Analyze Logs"</li> <li>• Results contain many images → widget offering a "View as Gallery" option.</li> </ul> <p>This module needs to analyze the query and the properties of the result set (file types, metadata aggregates, possibly even content snippets) to decide which widgets are relevant. This could range from simple rules (e.g., "if query contains 'image' and results include JPGs, show Gallery widget") to more complex analysis. Some applicable design patterns include:</p> <ul style="list-style-type: none"> <li>• <b>Factory:</b> A central <code>WidgetFactory</code> could manage available widgets and invoke their activation strategies based on the search context.</li> <li>• <b>Observer:</b> Widgets could potentially subscribe to search result updates to determine activation.</li> </ul>
9/10	<p><b>(Excellent):</b> Analyze the metadata of the documents within the current result set and display summary counts. Examples:</p> <ul style="list-style-type: none"> <li>• <b>File Type:</b> PDF (25), DOCX (12), TXT (8)</li> <li>• <b>Modified Year:</b> 2025 (20), 2024 (18)</li> <li>• <b>Language:</b> C (42), Java (666013 <b>G</b>)</li> </ul>
10/10	<p><b>(Outstanding):</b> Add a pre-commit <code>hook</code>  to your project (e.g., to run a linter or basic checks). Git <code>tag</code>  your final project version. Some popular versioning schemas include:</p> <ul style="list-style-type: none"> <li>• Semantic versioning</li> <li>• Calendar versioning</li> </ul> <p>Finally, document your progress in a <code>CHANGELOG.md</code><sup>1</sup> file.</p>

**⚠ Important: The deadline for this iteration is Week 13.**

<sup>1</sup><https://keepachangelog.com>. For example, `this`  is the React changelog.