

HEURISTIC 1

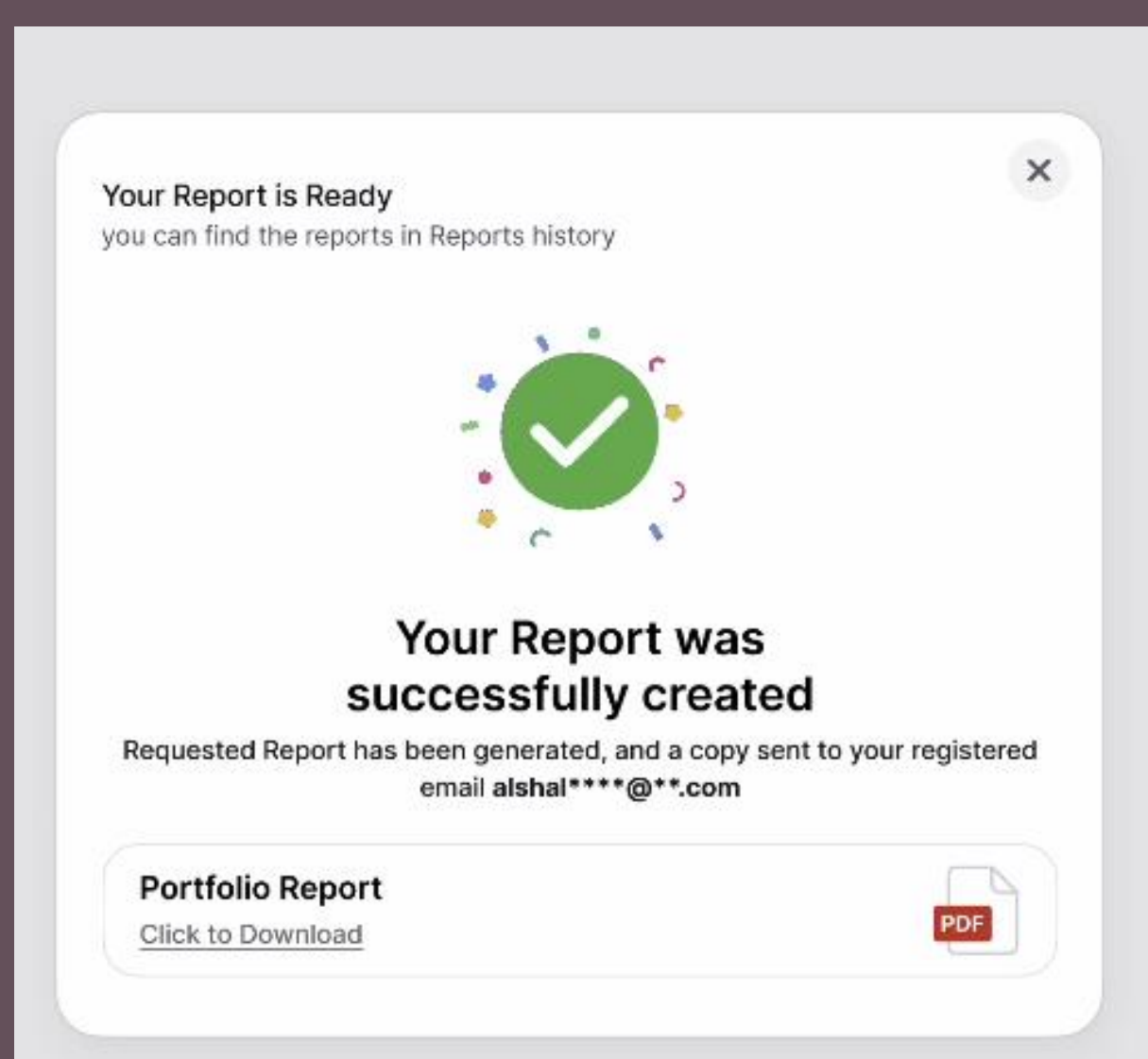
Visibility of System Status

The design should constantly **keep users informed about what is happening** through feedback mechanisms.



EXAMPLE

Utilize progress indicators to visually demonstrate ongoing processes



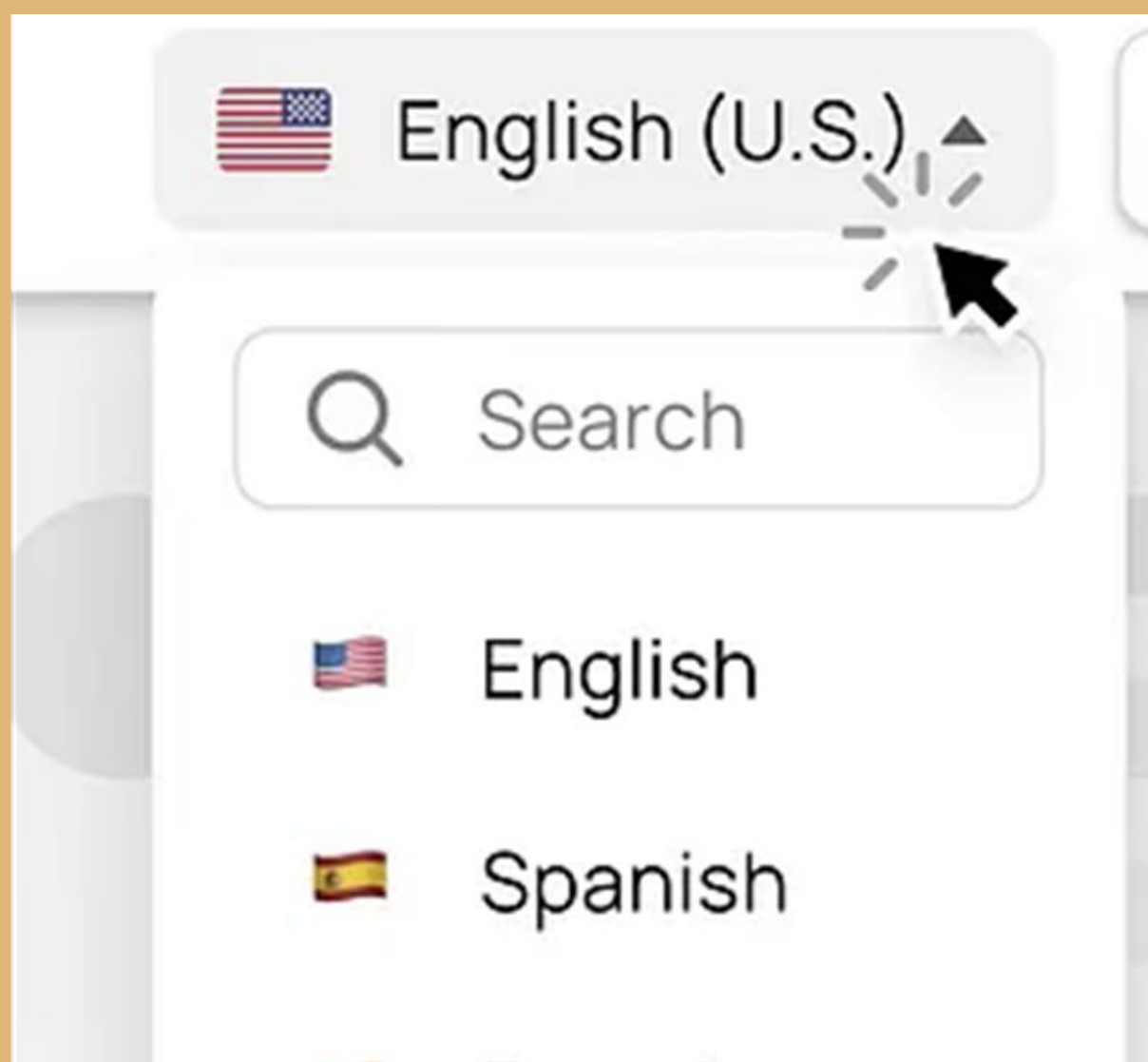
EXAMPLE

Implement notifications as direct feedback upon task completion to confirm successful actions.

HEURISTIC 2

Match between System & the Real World

Employ **language and symbols** that users are familiar with and reflect their experiences from the real world.



EXAMPLE

Optimize interface elements to reflect real-world conventions and analogies related to the context.

7.1 Introduction

A skill that a Games User Researcher should have is the ability to design quality methods and measures to answer questions about human behavior and attitudes with games. Games User Researchers should have a understanding of the limits of different methods and measures as well skill should be guiding teams to focus on the most appropriate questions because while all questions are interesting, not all are equally useful to a game better.

Picking or devising a method to answer questions is just like a sports skill-based endeavour. You get better at it the more you do it. So the try, the better you get (just like playing games). Engaging in method s

Games User Research, Anders Drachen, Pejman Mirza-Babaei, Lennart E. Nacke (Eds).
© Oxford University Press 2018. Published 2018 by Oxford University Press

EXAMPLE

Highlighting text in an e-reading app is similar to highlighting text in a book.

HEURISTIC 3

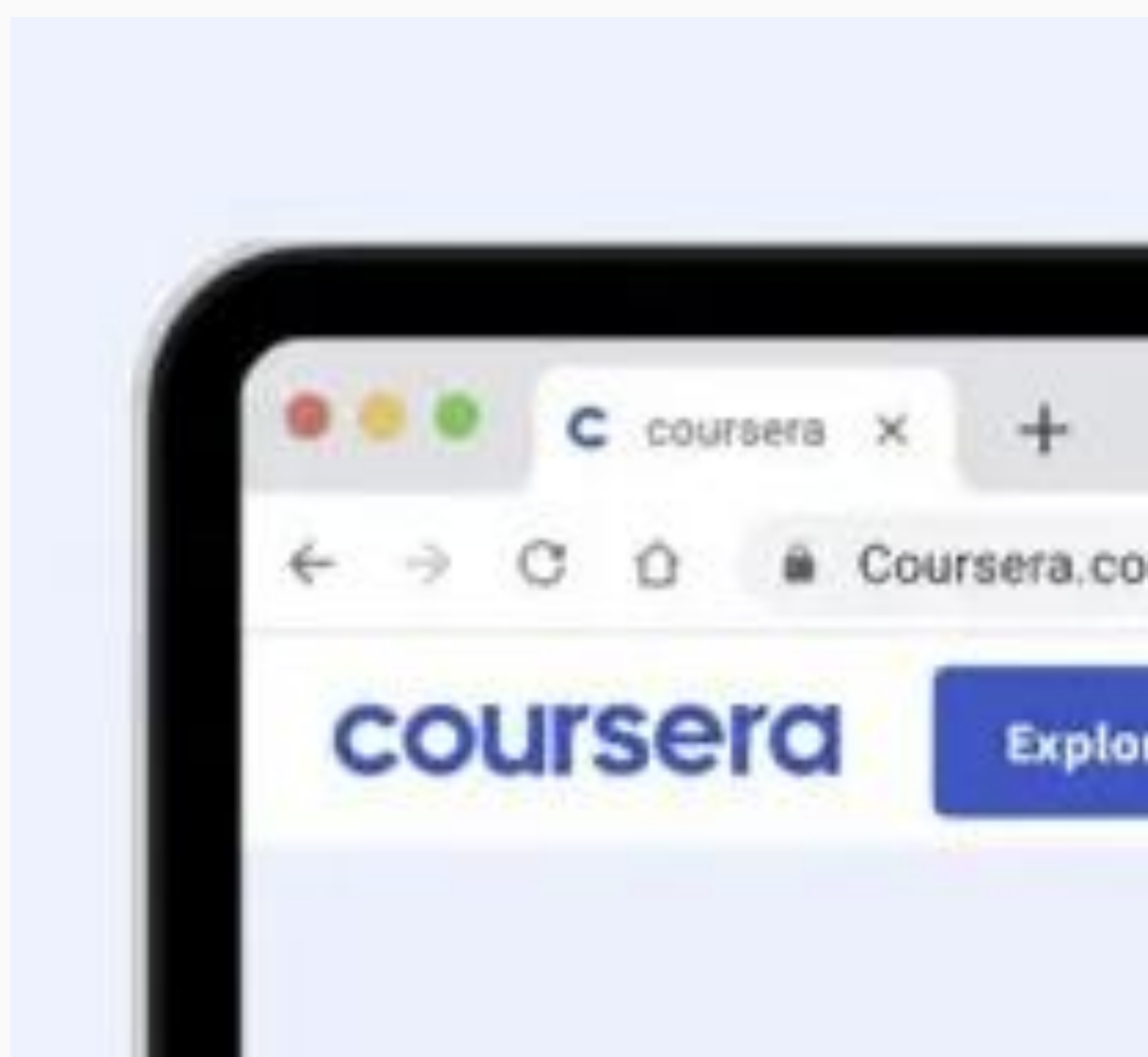
User Control & Freedom

Users often perform actions by mistake. They need a **clearly marked "emergency exit" to leave the unwanted action.**



EXAMPLE

Implement undo and redo functionalities to allow users to correct actions.



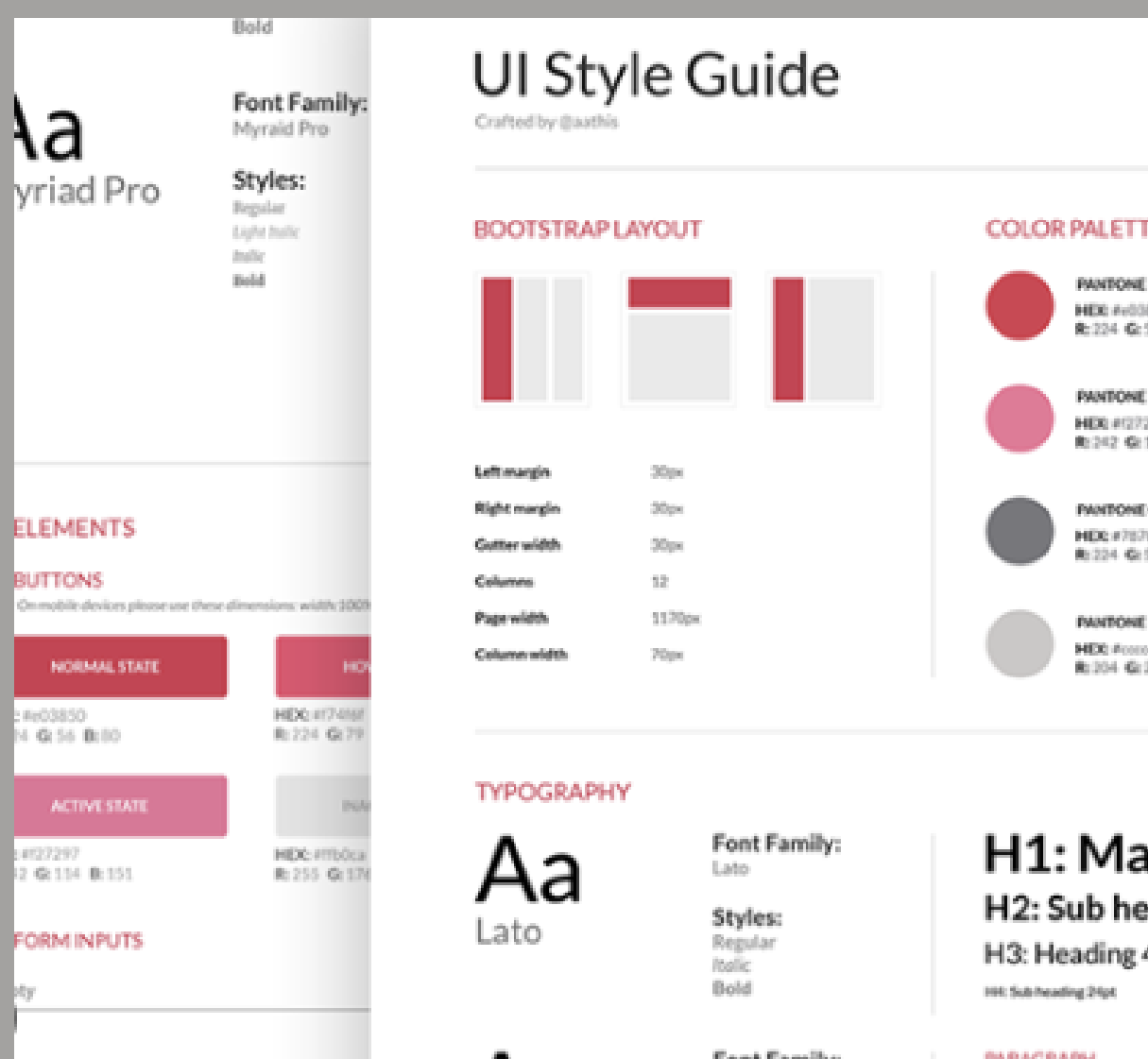
EXAMPLE

Make sure the exit is clearly labeled and discoverable.

HEURISTIC 4

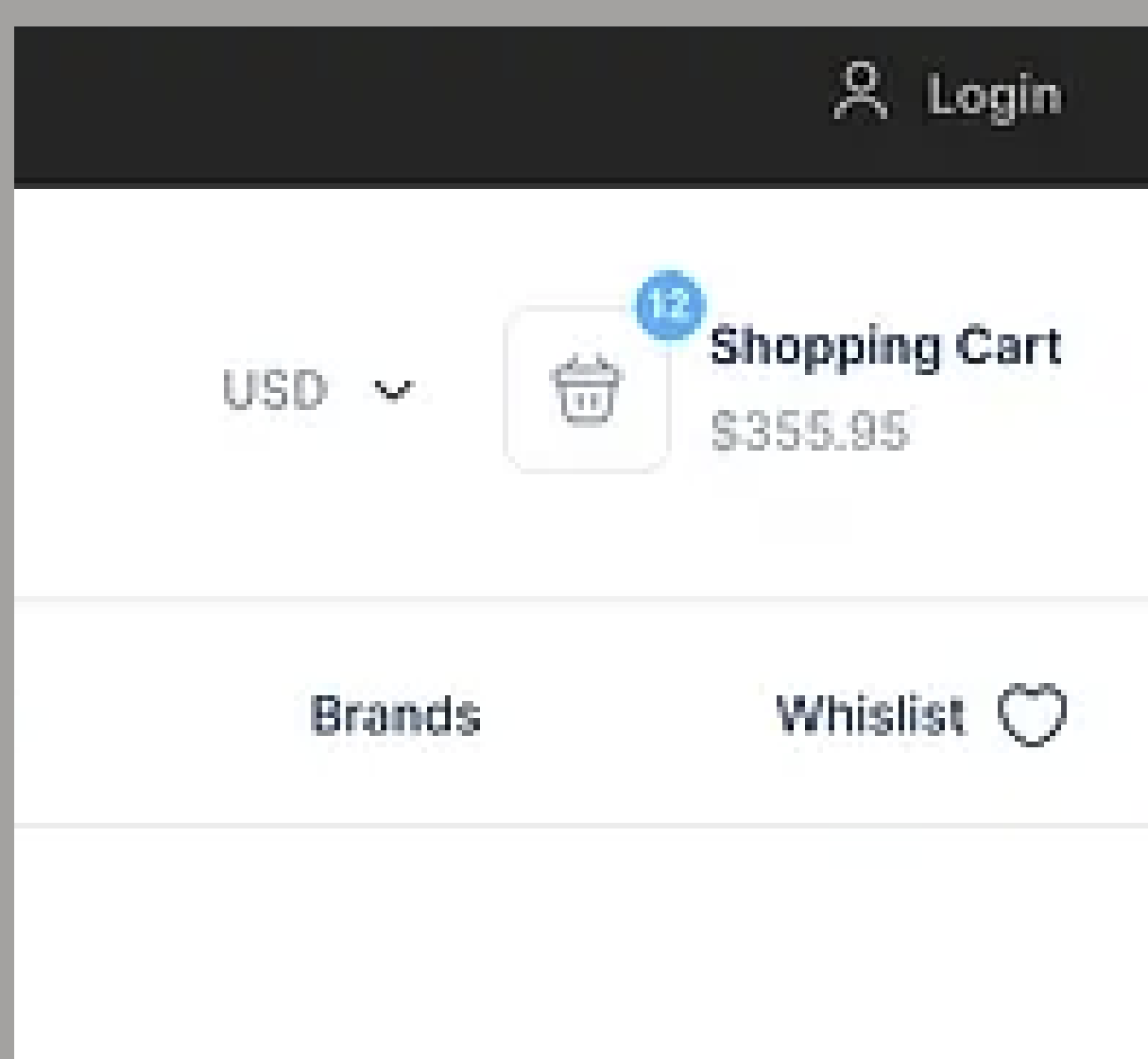
Consistency & Standards

The design should adhere to **consistent patterns and established conventions**.



EXAMPLE

Having a design system that ensures uniformity across products.



EXAMPLE

Aligning designs with industry standards to leverage existing user familiarity and expectations.

HEURISTIC 5

Error Prevention

Systems should **proactively prevent errors**, helping users **avoid mistakes** rather than just providing solutions afterward.



Create Password

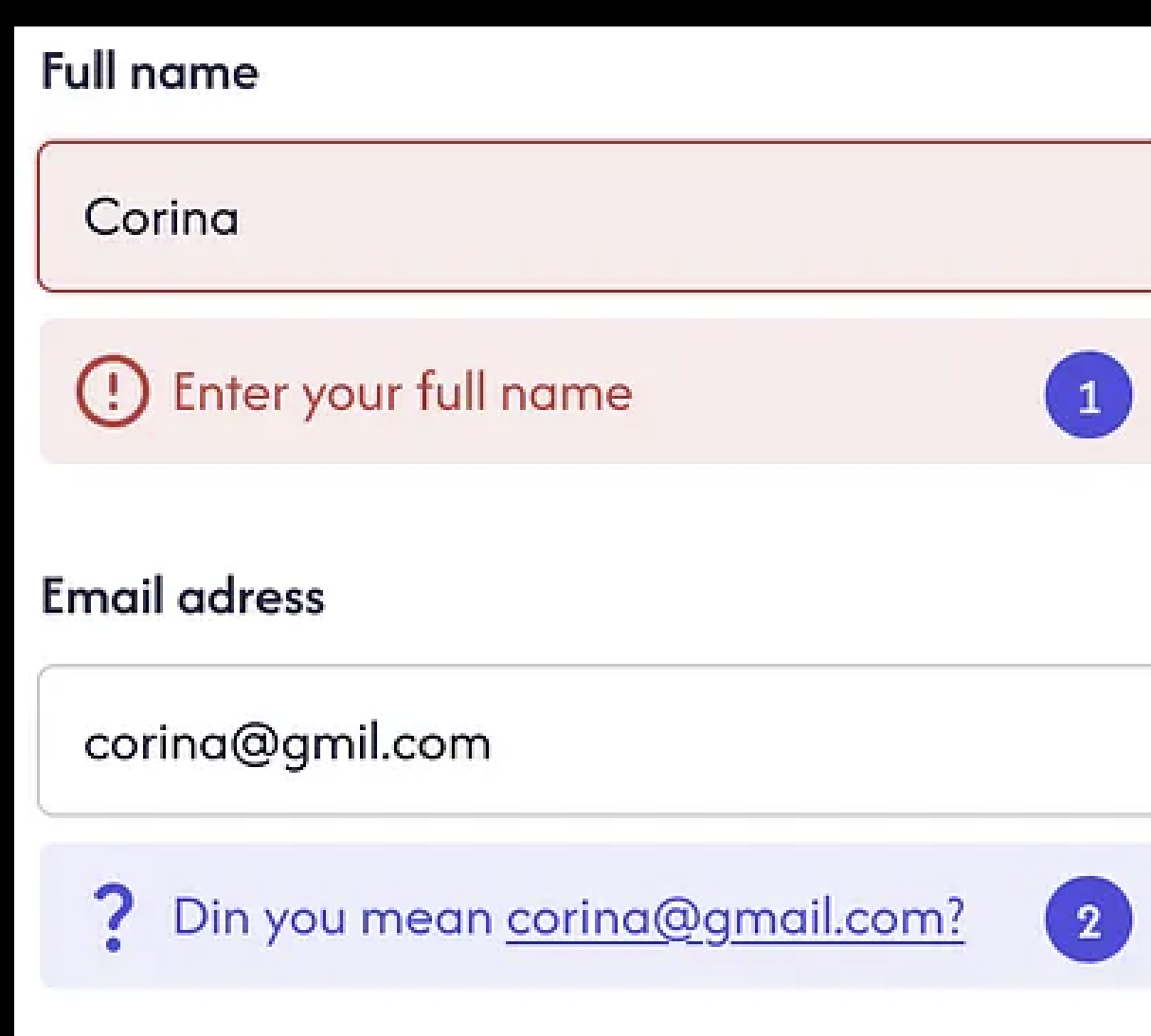
Password

Password must contain:

- ✓ At least 8 characters in length
- ✓ Lower case letters (a-z)
- ✓ Upper case letters (A-Z)
- ✗ Numbers (0-9)
- ✗ Special characters (!@#\$%&*)

EXAMPLE

Establish constraints and defaults that guide users seamlessly through tasks.



Full name

Corina

! Enter your full name 1

Email adress

corina@gmail.com

? Din you mean corina@gmail.com? 2

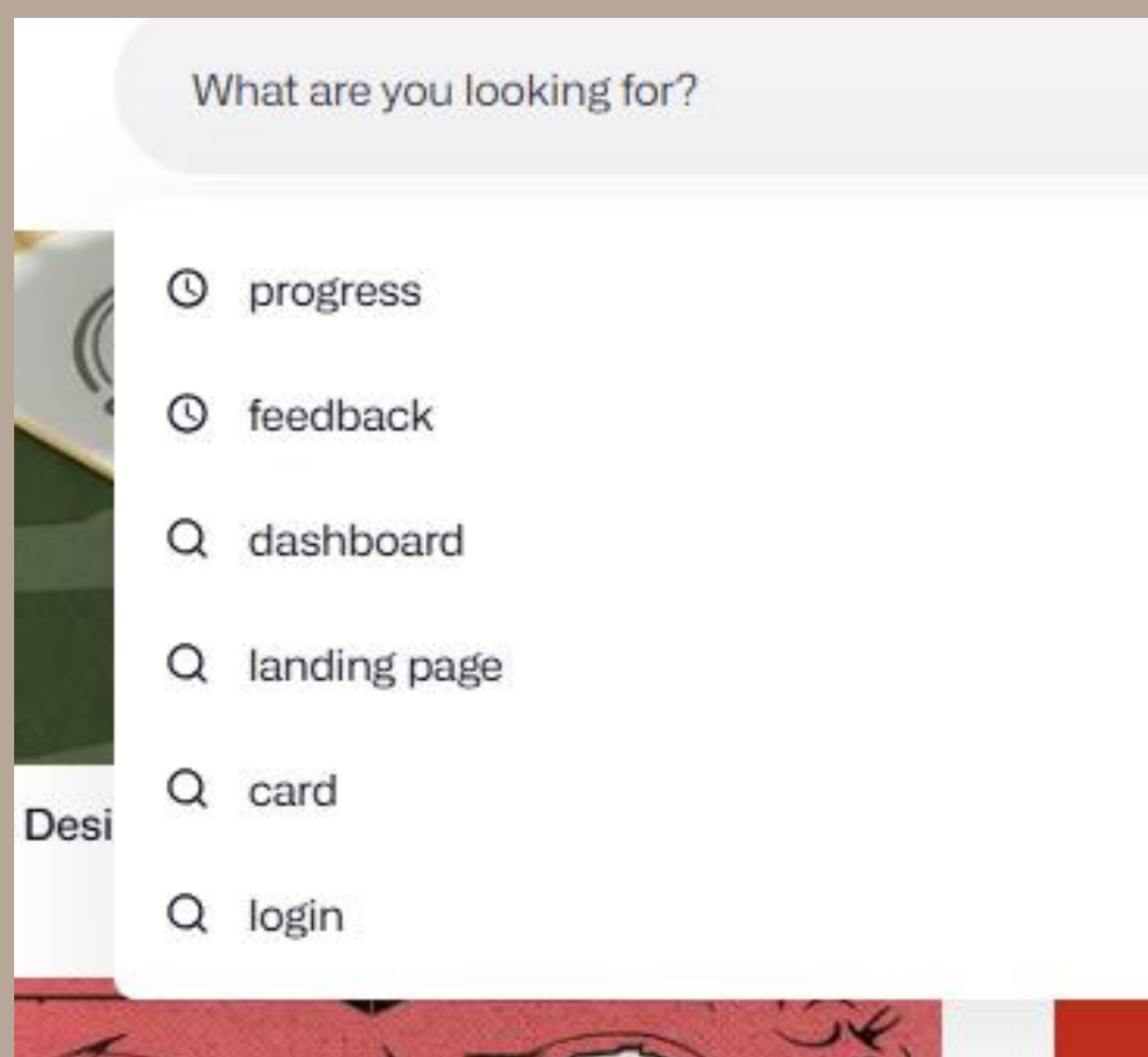
EXAMPLE

Prevent mistakes by removing memory burdens, supporting undo, and warning your users.

HEURISTIC 6

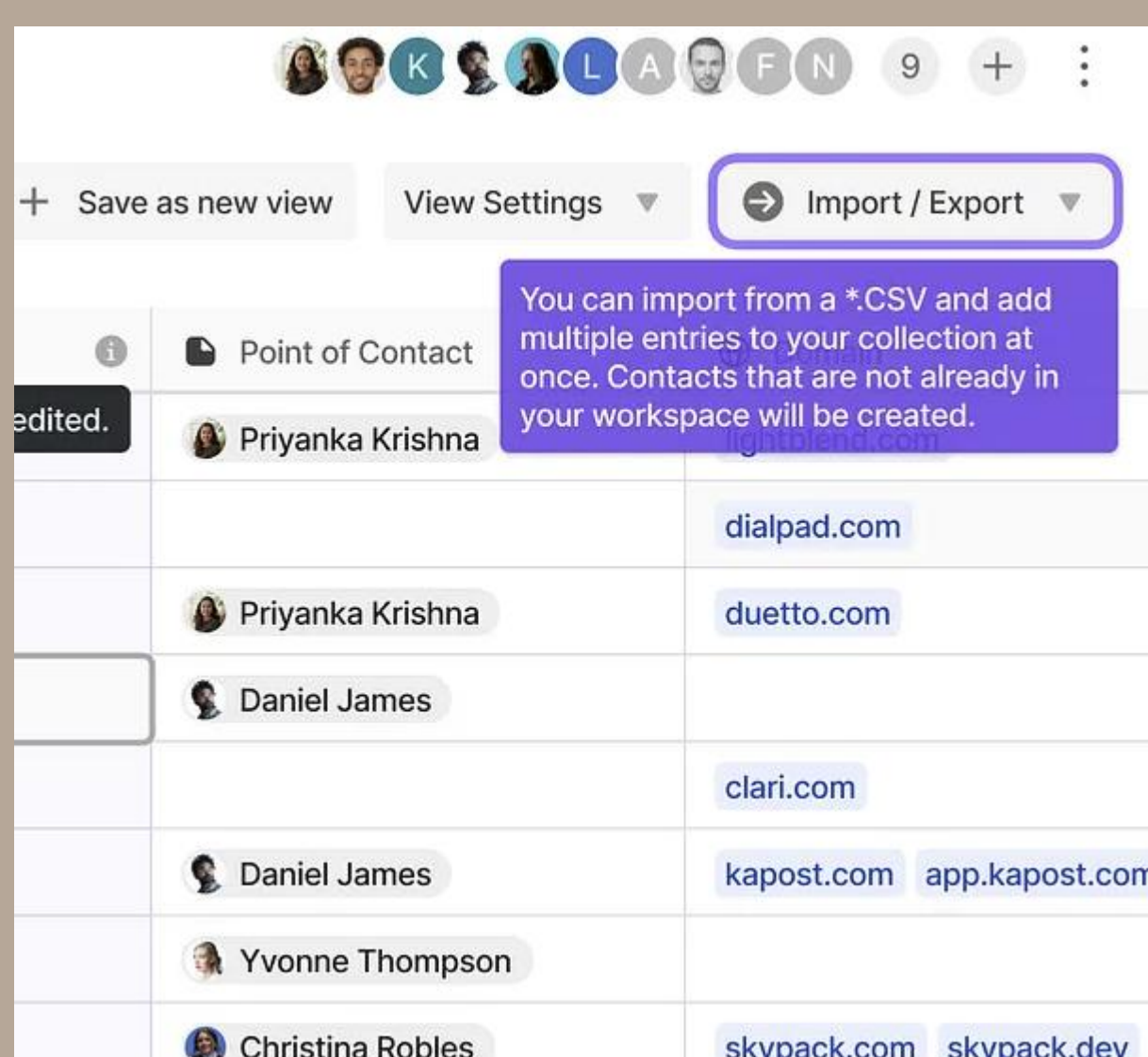
Recognition Rather Than Recall

The **system should rely on recognition of visible elements** rather than requiring users to recall information from memory.



EXAMPLE

Keeping frequently used options visible and easily accessible to users.



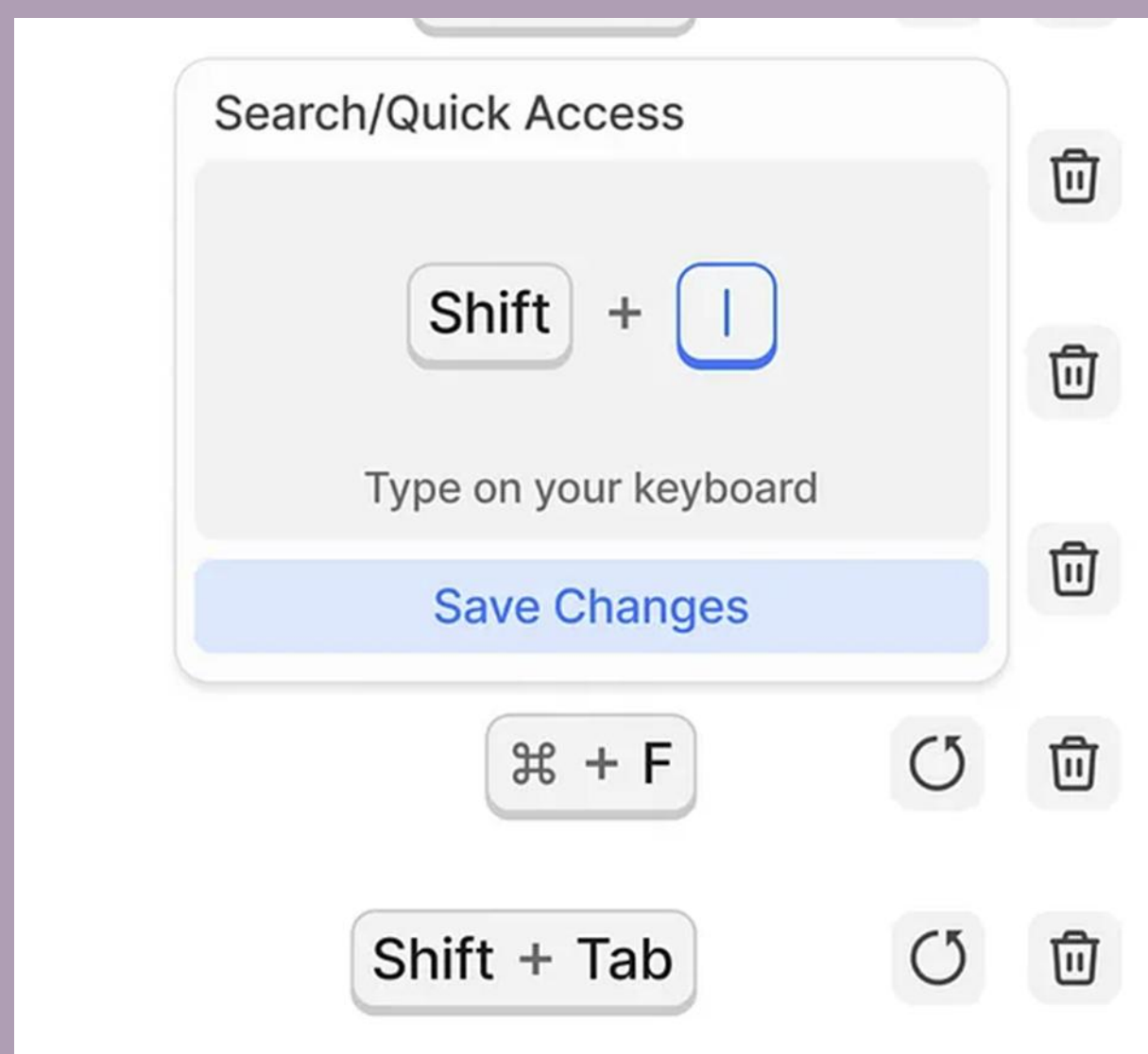
EXAMPLE

Offering contextual help through tooltips and hints, instead of giving users a long tutorial to memorize

HEURISTIC 7

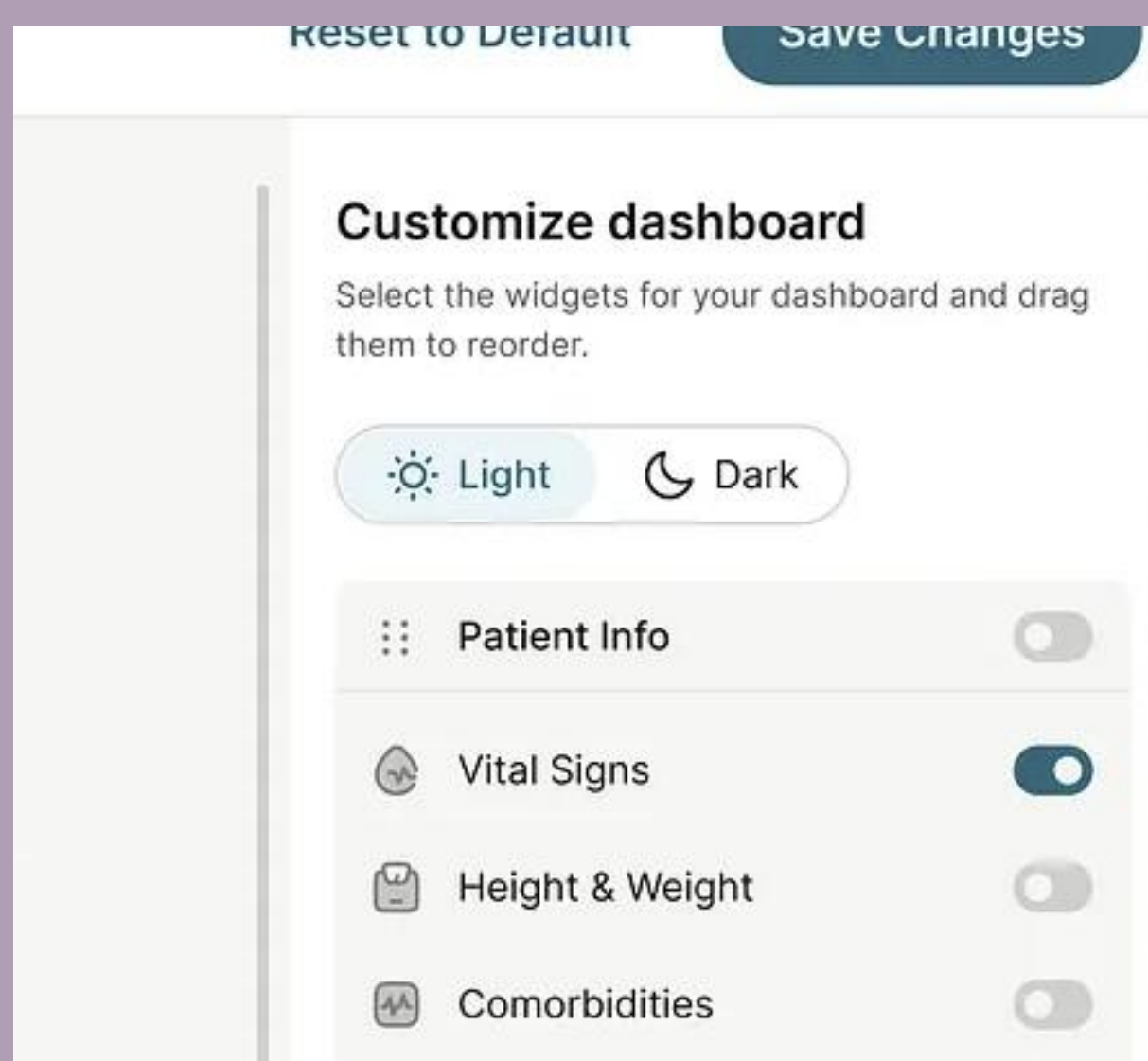
Flexibility & Efficiency of Use

Offer flexibility to accommodate both novice users and experts, providing shortcuts for efficiency.



EXAMPLE

Incorporating shortcuts and gesture controls to enhance task efficiency for advanced users.



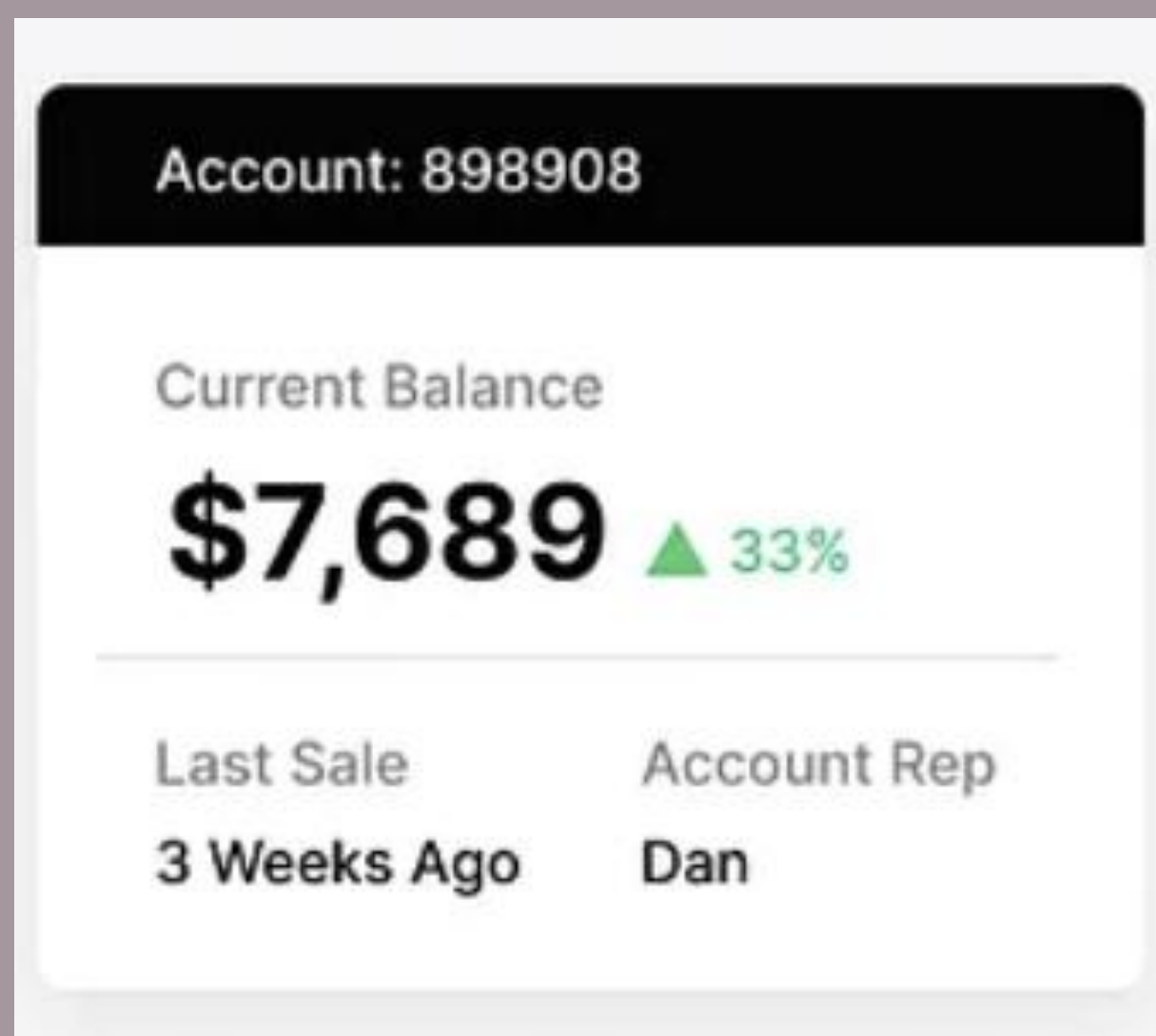
EXAMPLE

Allowing personalization options to tailor the interface to individual user workflows.

HEURISTIC 8

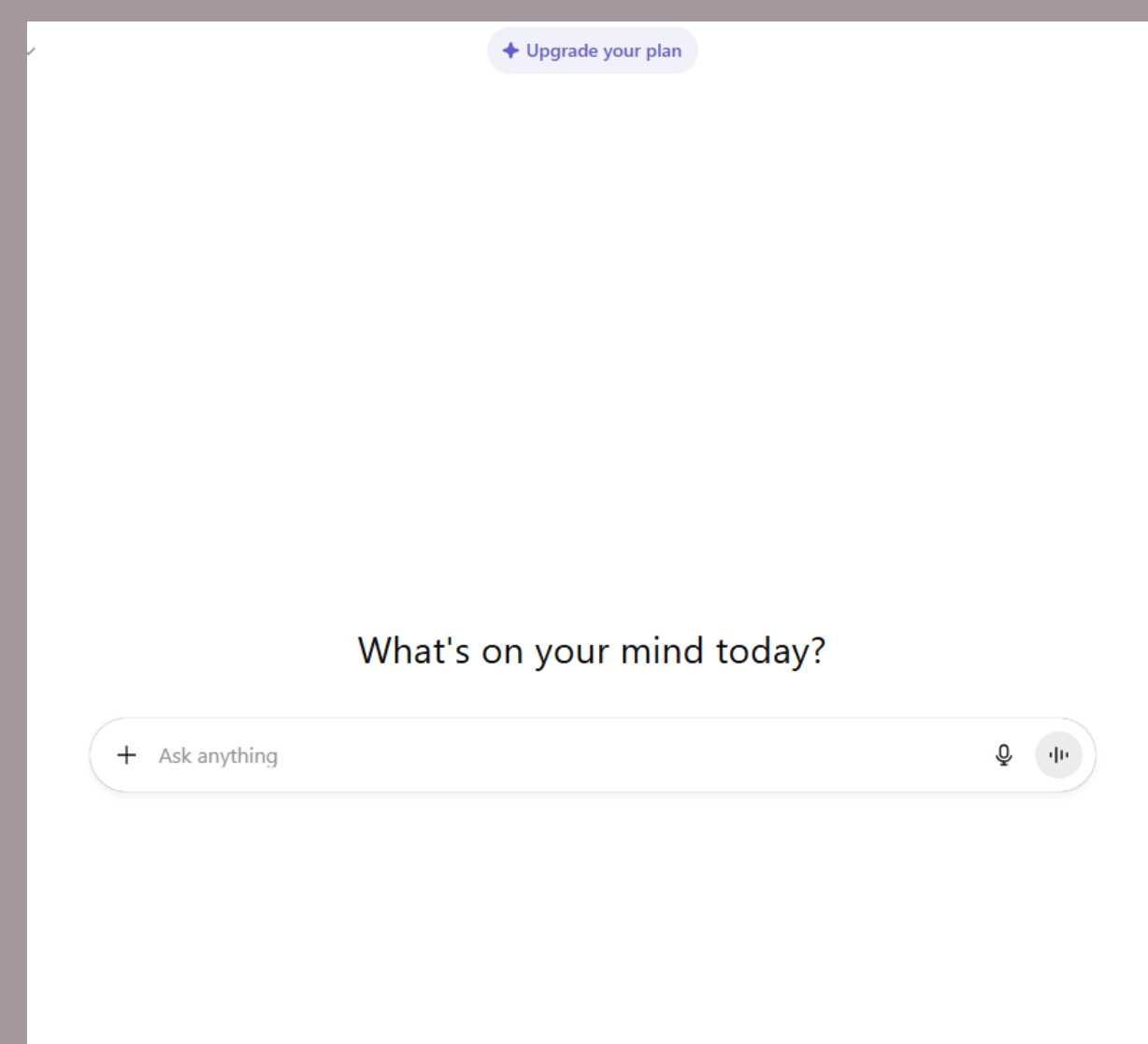
Aesthetic & Minimalistic Design

Interfaces should **not contain information that is irrelevant or rarely needed.**



EXAMPLE

Maintaining a clear visual hierarchy to direct user attention meaningfully



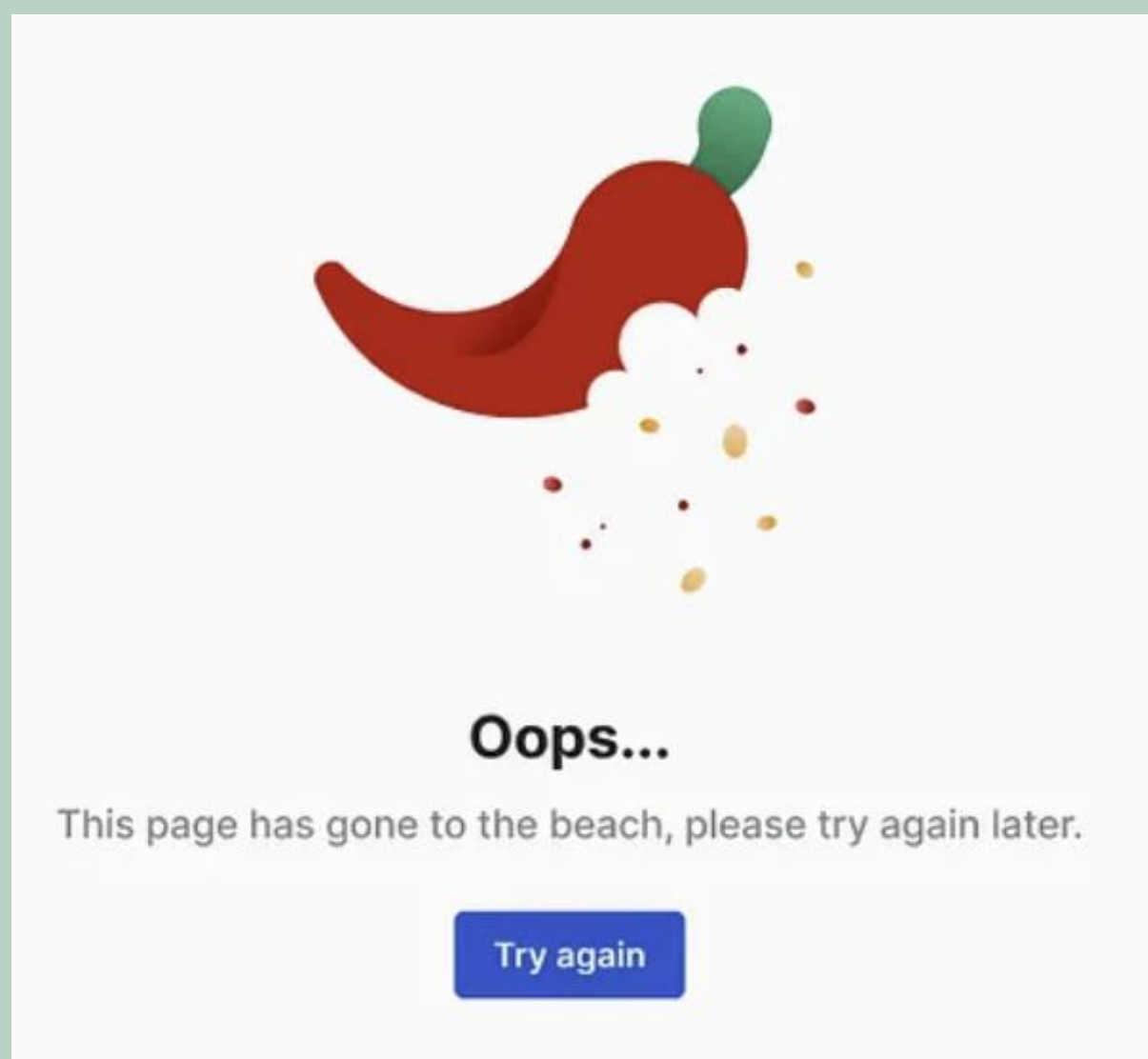
EXAMPLE

Prioritizing content and task-critical elements in the user-interface

HEURISTIC 9

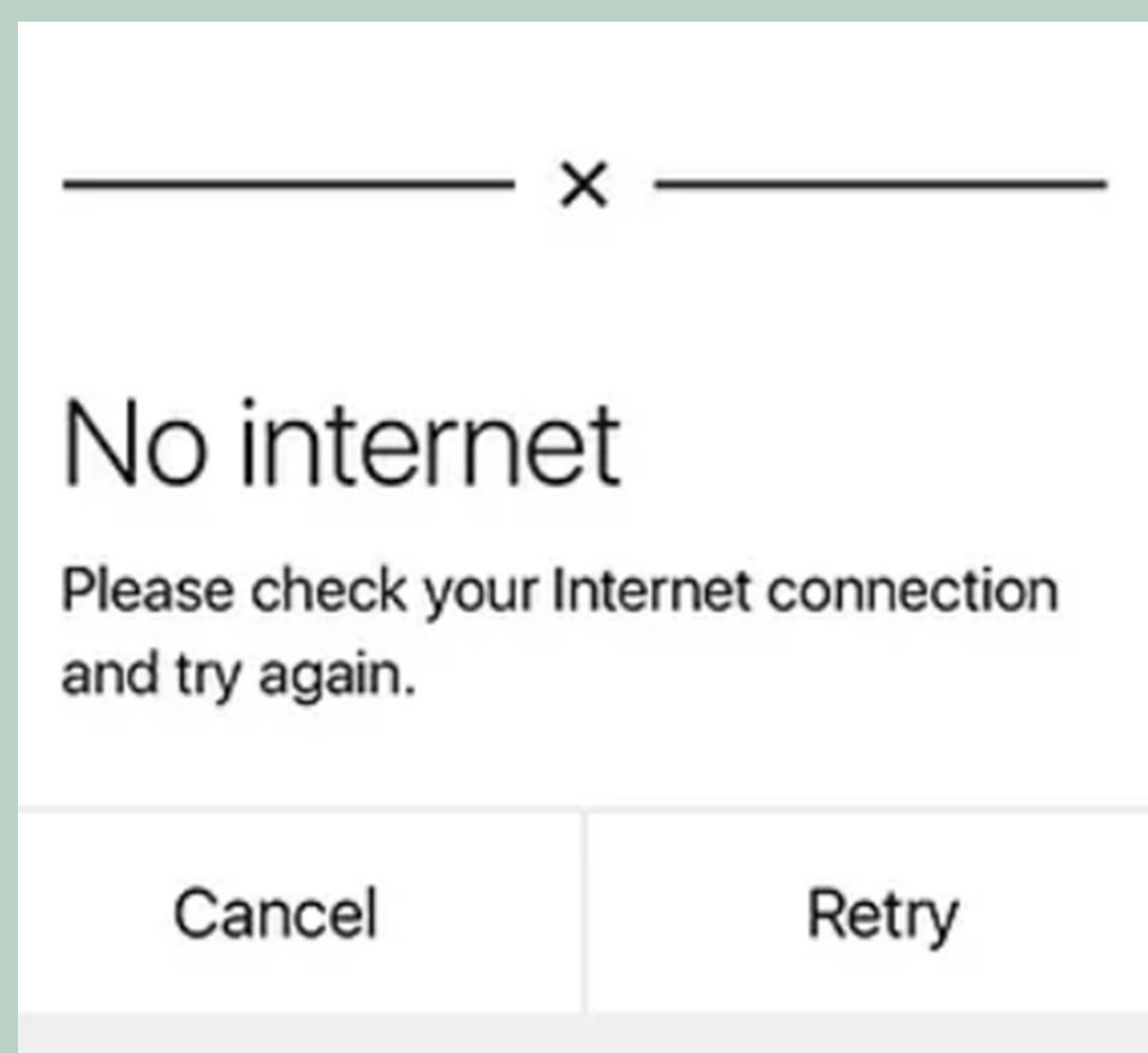
Help Recognize, Diagnose, & Recover from Errors

Error messages should **be expressed in plain language (no error codes)**, precisely indicate the problem.



EXAMPLE

Using simple language to describe errors clearly, avoiding technical jargon or error codes



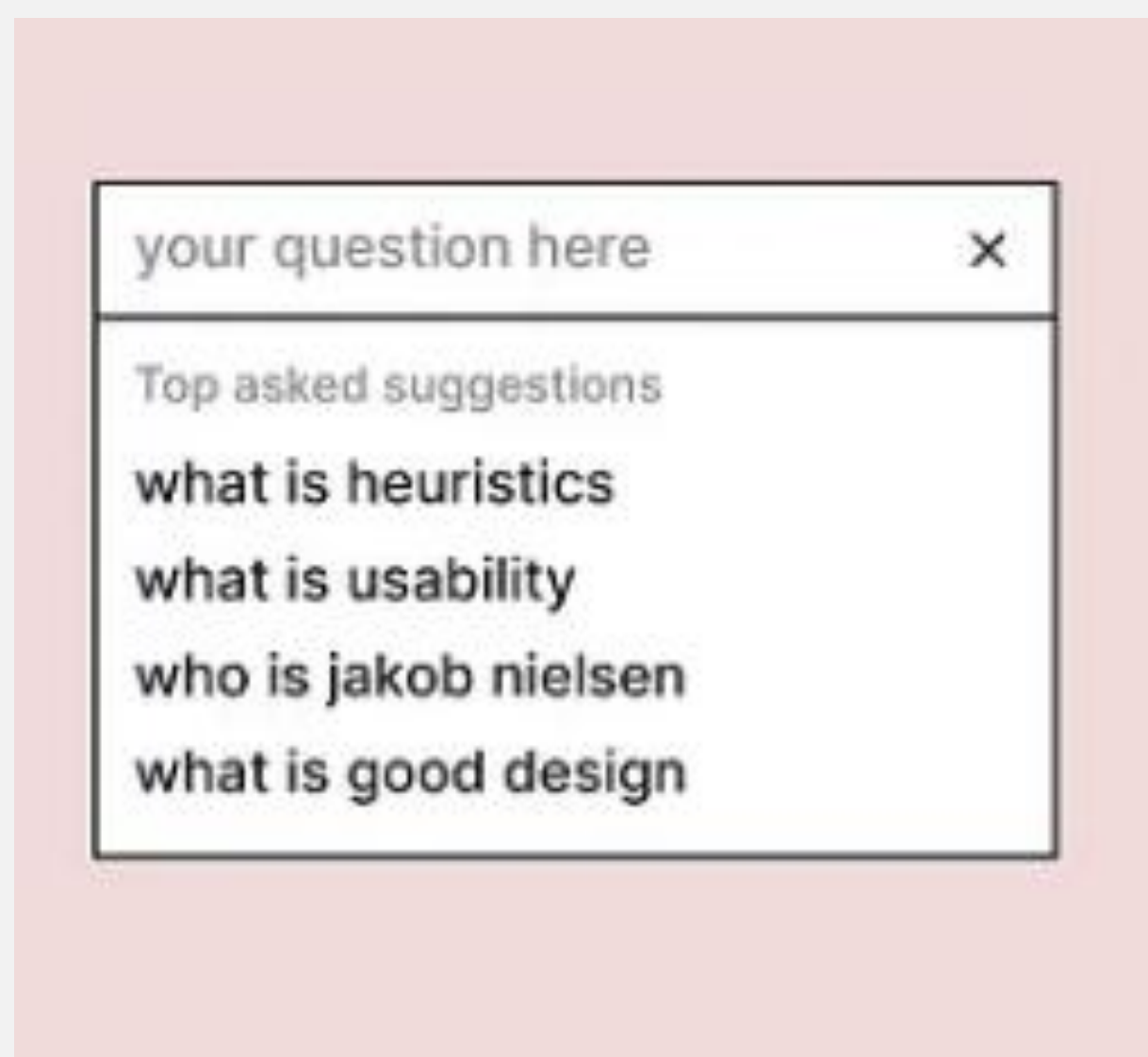
EXAMPLE

Offering suggestions or shortcuts to enable immediate error recovery.

HEURISTIC 10

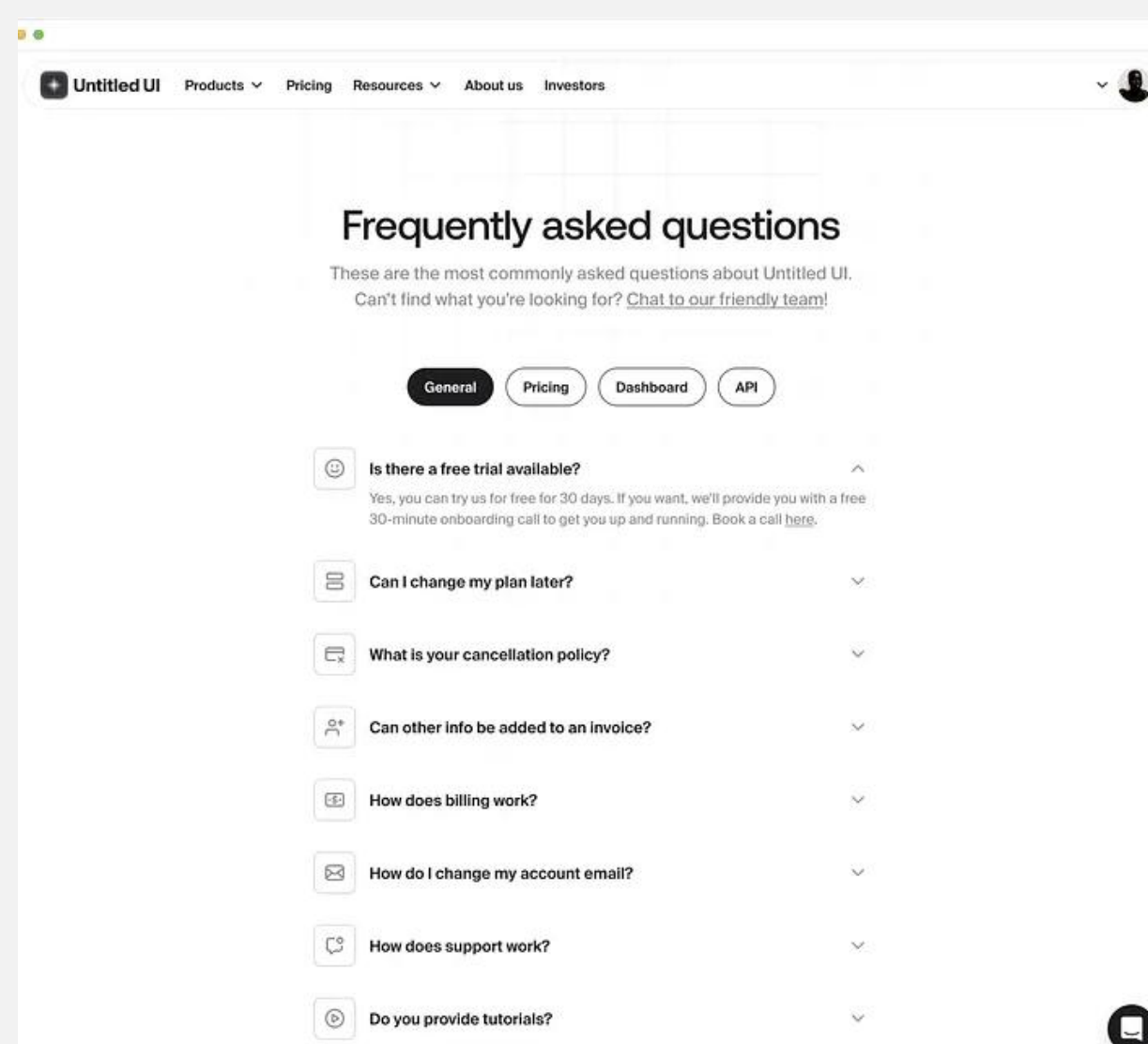
Help & Documentation

The design should **have accessible help documentation for guiding users** in understanding and completing tasks.



EXAMPLE

Having help documentation that is easily searchable and user-friendly.



EXAMPLE

Developing clear, concise FAQs and step-by-step guides to anticipate user questions and needs.