Remarkable Regular p1

Sparkle AI Website Generator: let Sparkle create a website for you with the power of AI in a few clicks

A new Figma-to-Sparkle plugin: import your Figma designs in Sparkle in seconds

Image/Video Gallery: a brand new component that easily lets you showcase your portfolio/media with an immersive lightbox experience

Grid Layout Block

New Interactive Elements: form slider, calculator smart field, date range options for popups

Remarkable RegRecog p1

Sparkle AI Website Generator: let sparkle create a website for you with the power of AI in a few clicks

A new Figma-to-Sparkle plugin: import your Figma designs in sparkle in seconds

Image/Video Gallery: a brand new component that easily lets you showcase your portfolio/media with an immersive light box experience

Grid Layout Block

New Interactive Elements: form slider, calculator smart field, date range options for popups

The Remarkable correctly combined my lines into single sentences; it missed a couple of capital "S"s and added a space (probably because "lightbox" is not a word).

I used the tablet's red color for the pen, so even though it only shows up as grey on the Remarkable, it displays as red on a device that can produce color, like my Mac.

Remarkable Regular p2

Hover Effects for Buttons

Text Effects: background clip and knockout

Box and Layout Block Blending Modes

Text Link Hover/Active/Visited States

Many performance improvements

Remarkable RegRecog p2

Hover Effects for Buttons

Text Effects: background clip and knockout

Box and Layout Block Blending Modes

Text Link Hover/Active/Visited States

Many performance improvements

No issues here

Remarkable List p1

A list of books that I've read recently.

Leigh Bardugo

The Grishaverse

- Shadow and Bone
- Siege and Storm
- Ruin and Rising

Peter F. Hamilton

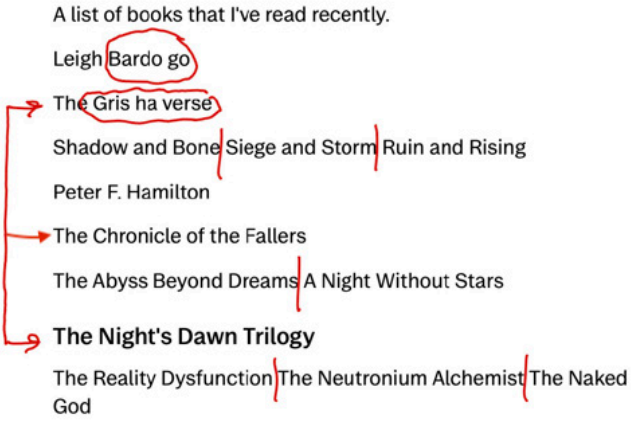
The Chronicle of the Fallers

- The Abyss Beyond Dreams
- A Night Without Stars

The Night's Dawn Trilogy

- The Reality Dysfunction
- The Neutronium Alchemist
- The Naked God

Remarkable ListRecog p1



The Remarkable added some spaces it shouldn't have, and its method of combining lines as if they were part of the sentence backfires.

Also note how it made the last line that I underlined bold, but not the first two.

Remarkable List p2

TABLET COMPARISONS

Kindle Pen with Eraser

- Pros
  - $70 cheaper than Remarkable option
  - Curved eraser head allow for more-focused erasing

- Cons
  - Weak magnet
  - Eraser head isn't stationary (compresses for some reason)
  - Gray color
  - Extraneous button

Remarkable ListRecog p2

TABLET COMPARISONS

Kindle Pen with Eraser

- Pros
- $70 cheaper than Remarkable option
- Curved eraser head allow for more- focused erasing
- Cons
- Weak magnet
- Eraser head isn't stationary (compresses for some reason)
- Gray color
- Extraneous button

Although the Remarkable correctly indented the first level of the list, it did not indent the second level. It also interpreted my dashes as dots.

It was unable to identify that "focused" should have attached to the "more-" despite combining the lines into one sentence.

Supernote Regular p1

Sparkle AI Website Generator: let Sparkle create a website for you with the power of AI in a few clicks

A new Figma-to-Sparkle plugin: import your Figma designs in Sparkle in Seconds

Image/Video Gallery: a brand new component that easily lets you showcase your portfolio/media with an immersive lightbox experience

Grid Layout Block

New Interactive Elements: form slider, calculator smart field, date range options for popups

Sparkle AI Website Generator: let
sparkle create a website for you with
the power of AI in a few clicks
A new Figma-to-sparkle plugin: import your
Figma designs in sparkle in seconds
Image/Video Gallery: a brand new component
that easily lets you showcase your
portfolio/media with an immersive
light box experience
Grid Layout Block
New Interactive Elements: form slider, calculator
Smart field, date range options for popups

The biggest problem with Supernote's recognition 'engine' is that it's not 'smart'. Each line of written words became its own line of text, breaking up all of the sentences.

It also didn't put extra vertical space between paragraphs like the Remarkable did.

Note that Supernote's highlighter/marker does not do colors. (I did the proofreading on the device itself.)

Supernote RegRecog p1

Supernote Regular p2

Hover Effects for Buttons

Text Effects: background clip and knockout

Box and Layout Block Blending Modes

Text Link Hover/Active/Visited States

Many performance improvements

Hover Effects for Buttons

Text Effects: background dip and knockout

Box and Layout Block Blending Modes

Text Link Hover/Active/Visited States

Many performance improvements

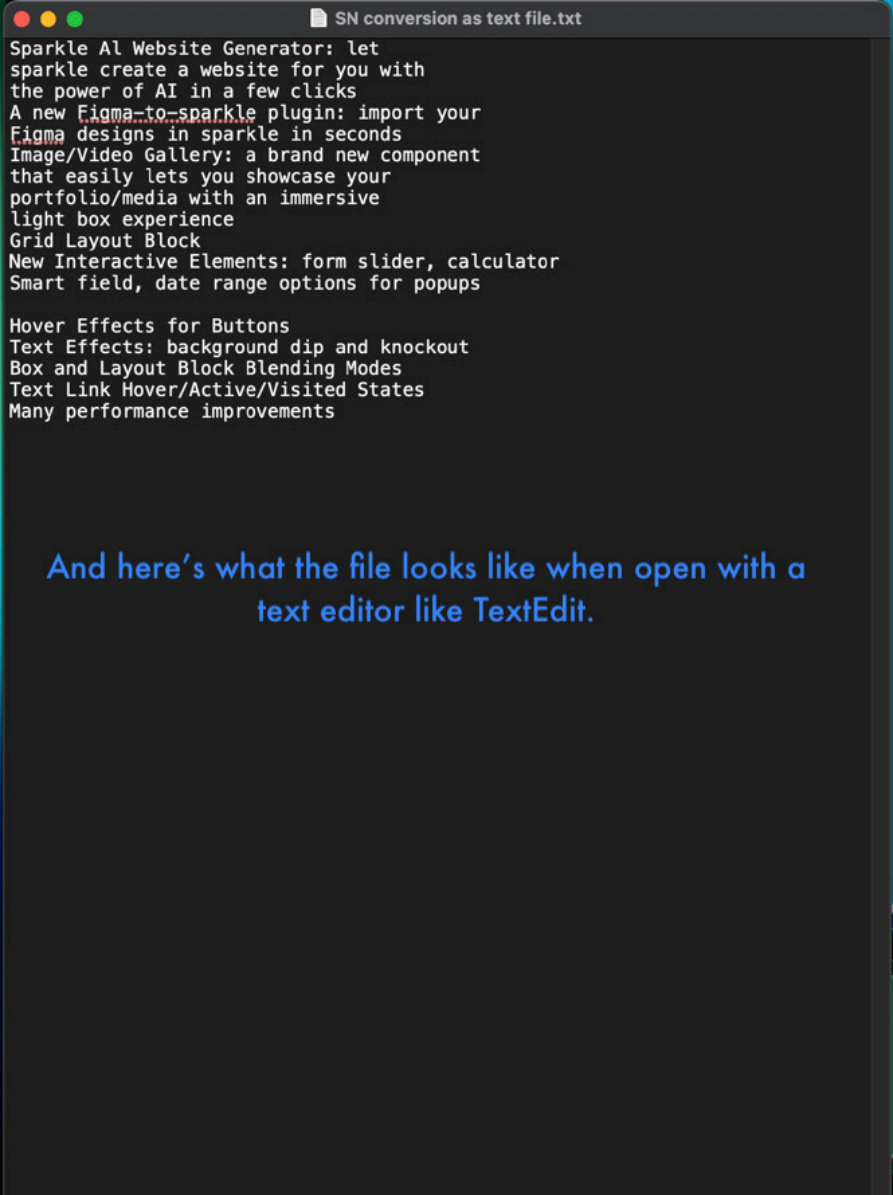
Unlike the Remarkable, the Supernote turned a "cl" into a "d" despite there being an obvious gap between the top of the "c" and the "l" in the written version.

Supernote RegRecog p2

Sparkle AI Website Generator: let
sparkle create a website for you with
the power of AI in a few clicks
A new Figma-to-sparkle plugin: import your
Figma designs in sparkle in seconds
Image/Video Gallery: a brand new component
that easily lets you showcase your
portfolio/media with an immersive
light box experience
Grid Layout Block
New Interactive Elements: form slider, calculator
Smart field, date range options for popups

Hover Effects for Buttons
Text Effects: background dip and knockout
Box and Layout Block Blending Modes
Text Link Hover/Active/Visited States
Many performance improvements

This image is just to show what it looks like when the Supernote exports its recognized text. Note how the lines are squished together further than when the device displays the interpreted text.


SN conversion as text file.txt
Sparkle Al Website Generator: let
sparkle create a website for you with
the power of AI in a few clicks
A new Figma-to-sparkle plugin: import your
Figma designs in sparkle in seconds
Image/Video Gallery: a brand new component
that easily lets you showcase your
portfolio/media with an immersive
light box experience
Grid Layout Block
New Interactive Elements: form slider, calculator
Smart field, date range options for popups
Hover Effects for Buttons
Text Effects: background dip and knockout
Box and Layout Block Blending Modes
Text Link Hover/Active/Visited States
Many performance improvements
And here's what the file looks like when open with a
text editor like TextEdit.

Sparkle AI Website Generator: let sparkle create a website for you with the power of AI in a few clicks A new Figma-to-sparkle plugin: import your Figma designs in sparkle in seconds Image/ Video Gallery: a brand new component that easily lets you showcase your portfolio/media with an immersive light box experience Grid Layout Block New Interactive Elements: form slider, calculator Smart field, date range options for popups Hover Effects for Buttons Text Effects: background dip and knockout Box and Layout Block Blending Modes Text Link Hover/ Active/Visited States Many performance improvements

When exporting the text conversion using the "reformat" option rather than the "original" option, Supernote wraps the lines as if they were sentences. However... it wraps EVERYTHING, with only a line break (and no separation) at the page split (between "popups" and "Hover")–so that the entire result is just a mess.

Still, when you have the file on your computer, it's probably easier to 1) click and 2) press Return once or twice wherever there should be a line break in this version than to 1) click, 2) backspace, 3) add a space, and then 4) add line spaces where applicable to properly arrange the result from the "original"-format export.

Supernote RegRecog - "reformat" option

Supernote List p1

A list of books that I've read recently.

Leigh Bardugo

The Grishaverse

Shadow and Bone
Siege and Storm
Ruin and Rising

Peter F. Hamilton

The Chronicle of the Fallers

The Abyss Beyond Dreams
A Night Without Stars

The Night's Dawn Trilogy

The Reality Dysfunction
The Neutronium Alchemist
The Naked God

A list of books that I've read recently.

Leigh Bardugo

The Gris traverse

Shadow and Bone

siege and storm

Ruin and Rising

Peter F. Hamilton

The Chronicle of the Fakers

The Abyss Beyond Dreams

A Night Without Stars

The Night's Dawn Trilogy

The Reality Dysfunction

The Neutron ivm Alchemist

The Naked God

Supernote made several letter-interpretation errors here, although the "k" in "Fallers" -> "Fakers" seems excusable given that I must've made a mark between the "l"s in my handwriting at the time.

Breaking up "Grishaverse" and "Neutronium" was inexplicable though. It also didn't capitalize that "S" in "Siege" despite it being the same size as the "S" in the previous handwritten line.

It also interpreted my "h" in "Grishaverse" as "tr."

Supernote ListRecog p1

Supernote List p2

TABLET COMPARISONS

Kindle Pen with Eraser

- Pros
  - $70 cheaper than Remarkable option
  - Curved eraser head allows for more-focused erasing

- Cons
  - Weak magnet
  - Eraser head isn't stationary (compresses for some reason)
  - Gray color
  - Extraneous button

TABLET COMPARISONS

Kindle Pen With Eraser

• Pros

- $70 cheaper than Remarkable option
- curved eraser head allows for more-focused erasing

e Con s

- weak magnet
- Eraser head isn't stationary (compresses for some reason)
- Gray color
- Extraneous button

Again, Supernote was unable to deduce where lines should be connected into a single sentence.

Despite correctly identifying my dashes in the sublists, it, again, didn't process the indenting because it just produces the lines of text regardless of location.

It also made an inexplicable blunder interpreting the dot in front of "Cons" as an "e." ?? That coupled with the gap between the "n" and "s" and the lowercasing of the obviously capitalized "W" in the subsequent line, definitely gave the Remarkable the win here.

Supernote ListRecog p2

A list of books that I've read recently.
Leigh Bardugo The Gris traverse Shadow and Bone siege and storm Ruin and Rising Peter F. Hamilton The Chronicle of the Fakers The Abyss Beyond Dreams A Night Without Stars The Night's Dawn Trilogy The Reality Dysfunction The Neutron ivm Alchemist The Naked God
TABLET COMPARISONS Kindle Pen With Eraser • Pros - $70 cheaper than Remarkable option - curved eraser head allows for more- focused erasing e Con s - weak magnet - Eraser head isn't stationary (compresses for some reason) - Gray color - Extraneous button

Again, Supernote combines lines everywhere when exporting text using the "reformat" option.

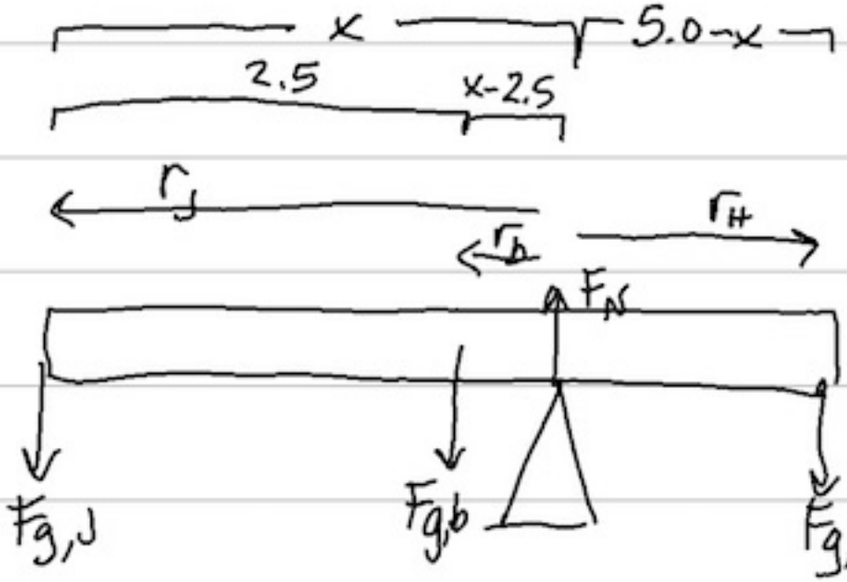
Somehow, the device places a line break after the first line (the only text not part of a list on that page), but not between lists. Again, there's a single line break and no separation between text from different pages of the note.

Supernote ListRecog - "reformat" option

Physics Calculus Ch 6 p234 #60

Remarkable Math/Science

Josie, who weighs 180 N, and 220-N Henry sit on opposite ends of a 5.00-m-long seesaw. The seesaw's plank has a weight of 250 N. What is the reaction force exerted by the seesaw's support, and where should the pivot be located if they are to be balanced?



"Infinite" pages–I had to crop the pdf to make the image similar to all the others here.

$\Sigma F = 0$; $\Sigma \tau = 0$ [conditions for equilibrium]

$\Sigma F = (-F_{g,J}) + (-F_{g,b}) + F_N + (-F_{g,H}) = 0$ [ ]

$(-180) + (-250) + F_N + (-220) = 0$

$F_N - 650 = 0$

determine force exerted

Physics Calculus Ch 6 p 234#60

Josie, who weighs 180N, and 220-N Henry sit on opposite ends of a 5.00-m- long seesaw. The seesaw's plank hasta weight of 250 N. What is the reaction force exerted by the seesaw's support, and where should the pivot be located if they are to be balanced?

(Ignore) This is from the diagram, which isn't a fair judge of OCR.

5. O-x

2.5

X-2.5

r,

rit

⇐ F

Fgb

Fg, J

As expected, the 'recognition' of the equations is pretty bad. In the main paragraph, Remarkable wasn't able to connect the across-lines "5.00-m-long" phrase properly, and couldn't interpret the necessary space between the "180" and the units "N" (though, to be fair, my handwriting didn't make the spacing clear at all).

Problematic translation of "has a" into "hasta" even though there's a giant gap in the handwriting.

Fg, ItEF-O, 85=0 (conditions for

equilibrium] E FIL-Fg,;) + f- Fg, b) + Fn + C- Fg,#)-0 (g) (-180) + (-250) + Fn + (-220) -O

determine force exerted by support

FN-650=0

FN-650N

[all forces are perpend.

Remarkable MathSciRecog

ET-ERF

to the moment-arms G. e. distance vectors) so 0=900 & sin O = I]E r F:(-r,. Fg,,) + (-rb. Fg, b) + (O. Fn) + (ra-Fg, a) o

[arbitrarily choosing torque out of screen as positive)

- I Fg,-r, Fg, btr, Fg,# = 0
- (x) (180)-(x-2. 5) (250) * (5.0-x) (220) = O
- 180×-250×+625+1100-2204=0

(There's a little more, but I cut it off to keep image size consistent.)

College Physics Ch 10 p349 #27

A steel bar is bent into a circular shape to produce a ring with a small gap (see figure). The width of the gap is 1.285 cm when the temperature is 42.0°C. Determine the width of the gap when the temperature is increased to 242.0°C. The coefficient of linear expansion for steel is $11 \times 10^{-6}\ °C^{-1}$



$$\Delta L = \alpha L_0 \Delta T$$

$$\Delta L = (11 \times 10^{-6})(1.285)(242-42)$$

$$= 2.83 \times 10^{-3}\ cm$$

[Imagine ring complete and analyze the 1.285-cm segment as if it were there]

$$L_{final} = L_0 + \Delta L$$

$$L_{final} = 1.285 + 2.83 \times 10^{-3}$$

$$L_{final} = 1.288\ cm$$

[Note that the gap increases]

College Physics Ch 10 p 349 #27
A steel bar is bent into a circular shape to produce a ring with a small gap (see figure). The width of the gap is 1.285cm when the temperature is 42.0°C. Determine the width of the gap when the temperature is increased to 242.0°C. The coefficient of linear expansion for steel is @ 11×10-6 ◦ c- I
DL = ✗ ↳ DT Imagine ring complete
DL = (11×10-6) (1.285) (242-42) and analyze the
= 2.83 ✗ 103cm 1.285-cm segment as
if it were there]
[final = Lot DL
(final = 1.285 + 2.83×10-3
[Note that the gap
increases
(final = 1.288cm

Some of the equation interpretation is pretty accurate, but it would still need a lot of editing to make complete sense of it because I organized that part of the note as two columns.

Like the Remarkable, the Supernote had difficulty understanding the desired separation of numbers and units.

Supernote MathSciRecog

Remarkable - ePUB

Parents and children

At the core of SwiftUI is its three-step layout process:

1. A parent view proposes a size for its child.
2. Based on that information, the child then chooses its own size and the parent view must respect that choice.
3. The parent view then positions the child in its coordinate space.

It sounds so trivial, and you're probably wondering why I'm starting out by mentioning something that is so straightforward, but the simple truth is that this simple process unlocks a huge amount of power and the more you really understand it the more you'll be able to get SwiftUI doing exactly what you want.

The key to the power is answering a simple question: what is the "parent view" in that process? When you're learning SwiftUI, the answer seems obvious. For example:

```
VStack {
   Text("Hello, world!")
      .frame(width: 300, height: 100)

   Image(systemName: "person")
}
```

The Remarkable wasn't able to differentiate different fonts in the ePUB (see SN image for what it should look like).

If I asked a learner what the parent of the text view was, they

Remarkable - ePUB

that lets you type to search through your code and also Apple's own APIs; activate it using Shift+Cmd+O, then type ModifiedContent and press return. This will open Xcode's generated interface file for SwiftUI, and you should be able to find this in there:

```
@frozen public struct ModifiedContent<Content, Modifier>
```

A little further down you'll also see its View conformance:

```
extension ModifiedContent : View where Content : View, Modifier : Vie
```

These things aren't magic, and they aren't *secret* – you can use ModifiedContent directly if you want, because it's public API.

For example, back in ContentView.swift we could create a custom a modifier like this one:

Note the tight spacing between the lines throughout the page.

```
struct CustomFont: ViewModifier {
    func body(content: Content) -> some View {
        content.font(.largeTitle)
    }
}
```

We could apply that to some text using ModifiedContent, like so:

```
struct ContentView: View {
    var body: some View {
        ModifiedContent(content: Text("Hello"), modifier: CustomFont())
    }
```

Supernote - ePUB

Some fonts that the Supernote uses had issues: here the apostrophes are disasters.

Parents and children

At the core of SwiftUI is its three-step layout process:

1. A parent view proposes a size for its child.
2. Based on that information, the child then chooses its own size and the parent view must respect that choice.
3. The parent view then positions the child in its coordinate space.

It sounds so trivial, and you’re probably wondering why I’m starting out by mentioning something that is so straightforward, but the simple truth is that this simple process unlocks a huge amount of power and the more you really understand it the more you’ll be able to get SwiftUI doing exactly what you want.

The key to the power is answering a simple question: what is the “parent view” in that process? When you’re learning SwiftUI, the answer seems obvious. For example:

```
VStack {
 Text("Hello, world!")
                    300, height: 100
```

Supernote - ePUB

```
    struct CustomFont: ViewModifier {
     func body(content: Content) -> some View {
         content.font(.largeTitle)
     }
}
```

...but Supernote was able to differentiate the fonts and present the text as such, with proper spacing between paragraphs and code blocks.

We could apply that to some text using `ModifiedContent`, like so:

```
    struct ContentView: View {
     var body: some View {
         ModifiedContent(content: Text("Hello"), modi
     }
}
```

That's obviously a lot more wordy than the normal SwiftUI code we'd write, but what I want you to understand is that the end result is identical to what we'd get by using `modifier()` like this:

```
    Text("Hello")
     .modifier(CustomFont())
     .onTapGesture {
         print(type(of: self.body))
```

Supernote - ePUB

This font remedied the the apostrophes, but the list indentation was still an issue.

Parents and children

At the core of SwiftUI is its three-step layout process:

1. A parent view proposes a size for its child.
2. Based on that information, the child then chooses its own size and the parent view must respect that choice.
3. The parent view then positions the child in its coordinate space.

It sounds so trivial, and you're probably wondering why I'm starting out by mentioning something that is so straightforward, but the simple truth is that this simple process unlocks a huge amount of power and the more you really understand it the more you'll be able to get SwiftUI doing exactly what you want.

The key to the power is answering a simple question: what is the "parent view" in that process? When you're learning SwiftUI, the answer seems obvious. For example:

```
VStack {
    Text("Hello, world!")
        .frame(width: 300, height: 100)

    Image(systemName: "person")
}
```

Supernote - ePUB

Same screen as previous... after I moved the toobar to the left side: note how the Supernote did not reformat the ePUB text to fit beside it (despite the toolbar not overlapping the text when the toolbar started on the left (see 2 and 3 images back).

At the core of SwiftUI is its three-step layout process:

1. A parent view proposes a size for its child.
2. Based on that information, the child then chooses its own size and the parent view must respect that choice.
3. The parent view then positions the child in its coordinate space.

It sounds so trivial, and you're probably wondering why I'm arting out by mentioning something that is so straightforward, but he simple truth is that this simple process unlocks a huge amount of ower and the more you really understand it the more you'll be able o get SwiftUI doing exactly what you want.

The key to the power is answering a simple question: what is the parent view" in that process? When you're learning SwiftUI, the nswer seems obvious. For example:

```
VStack {
    Text("Hello, world!")
        .frame(width: 300, height: 100)

    Image(systemName: "person")
```

For comparison, here's what it looks like in the Books app on an iPad.

Parents and children

At the core of SwiftUI is its three-step layout process:

1. A parent view proposes a size for its child.
2. Based on that information, the child then chooses its own size and the parent view must respect that choice.
3. The parent view then positions the child in its coordinate space.

It sounds so trivial, and you're probably wondering why I'm starting out by mentioning something that is so straightforward, but the simple truth is that this simple process unlocks a huge amount of power and the more you really understand it the more you'll be able to get SwiftUI doing exactly what you want.

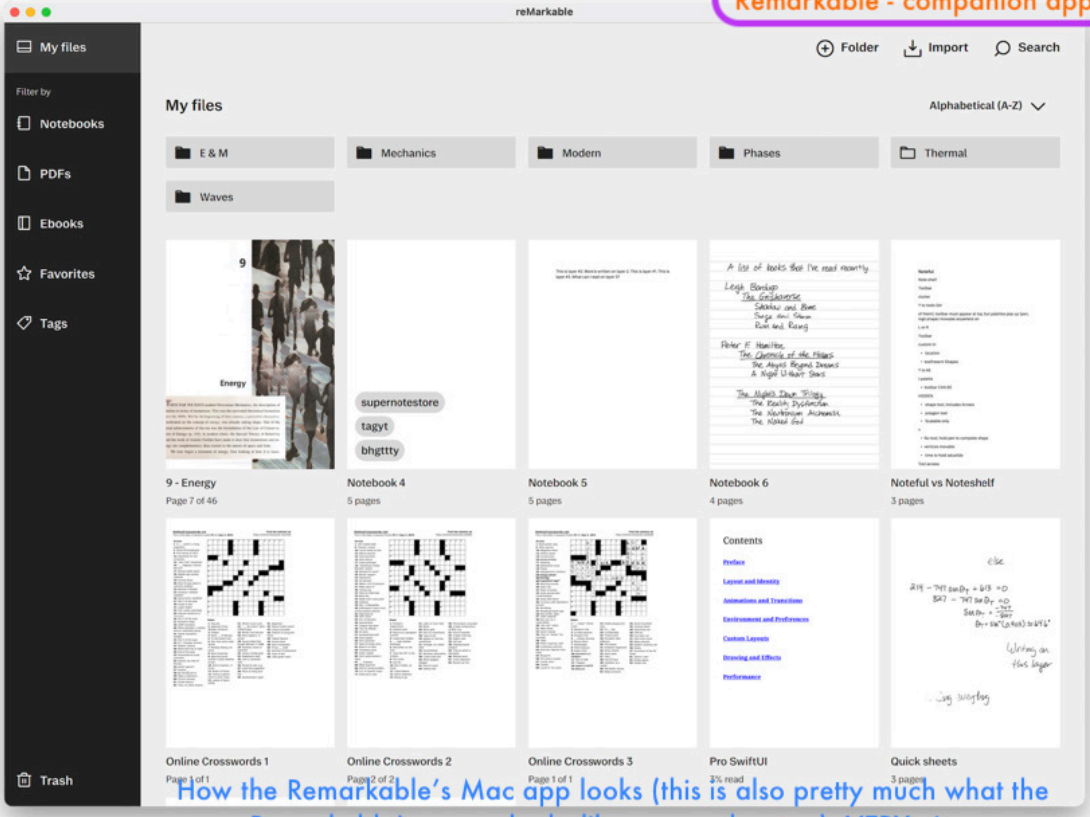
The key to the power is answering a simple question: what is the "parent view" in that process? When you're learning SwiftUI, the answer seems obvious. For example:

```
VStack {
    Text("Hello, world!")
        .frame(width: 300, height: 100)

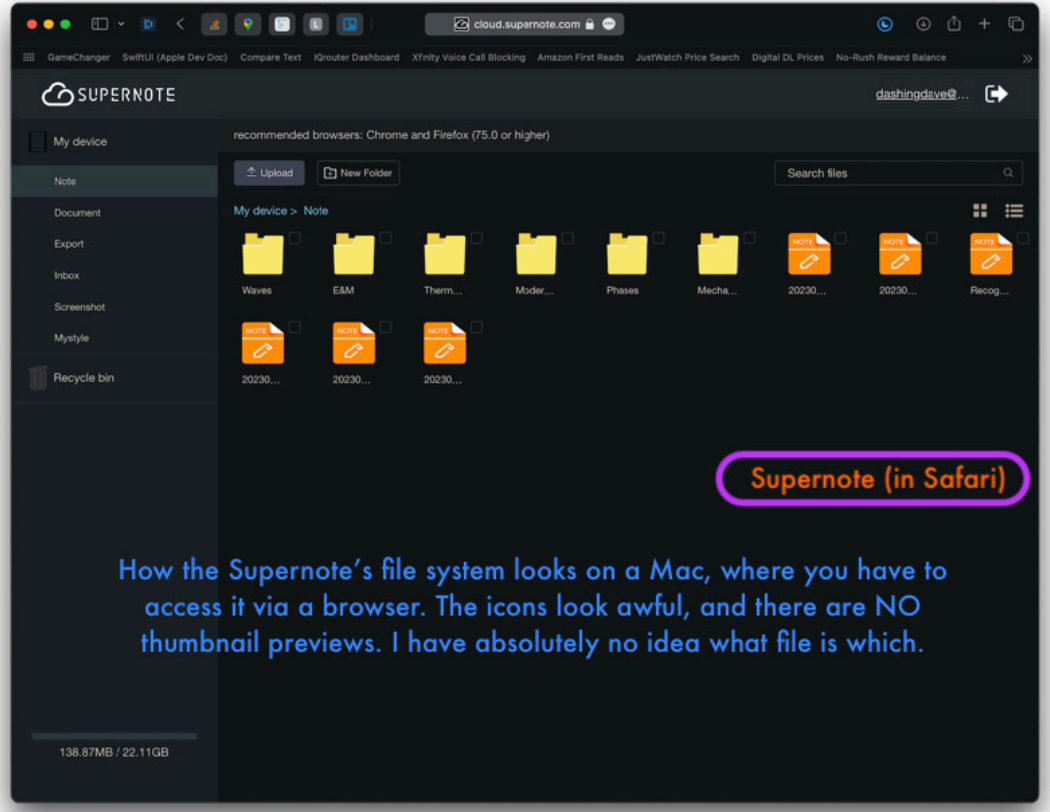
    Image(systemName: "person")
}
```

If I asked a learner what the parent of the text view was, they would probably answer "the **VStack**." And honestly that's a perfectly fine answer, and I wouldn't correct a beginner who said it – it feels natural, and it fits the hierarchy we can see when the code runs. However, it's also *wrong*, and once you have sufficient experience with SwiftUI it's important you understand why the answer is wrong, and more importantly what is *right*.

Remarkable - companion app



How the Remarkable's Mac app looks (this is also pretty much what the Remarkable's screen looks like–sans color–too). VERY nice.



How the Supernote's file system looks on a Mac, where you have to access it via a browser. The icons look awful, and there are NO thumbnail previews. I have absolutely no idea what file is which.

6:39 PM Sun Sep 24

92%

My files

Remarkable - iOS app

Alphabetical (A-Z)

E & M | Mechanics | Modern | Phases | Thermal | Waves



9 - Energy
Page 7 of 46



Notebook 4
5 pages



Notebook 5
5 pages



Notebook 6
4 pages



Noteful vs Noteshelf
3 pages



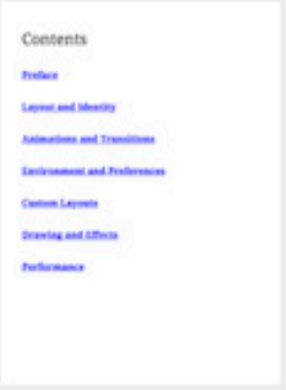
Online Crosswords 1
Page 1 of 1



Online Crosswords 2
Page 2 of 2



Online Crosswords 3
Page 1 of 1



Pro SwiftUI
3% read



Quick sheets
3 pages

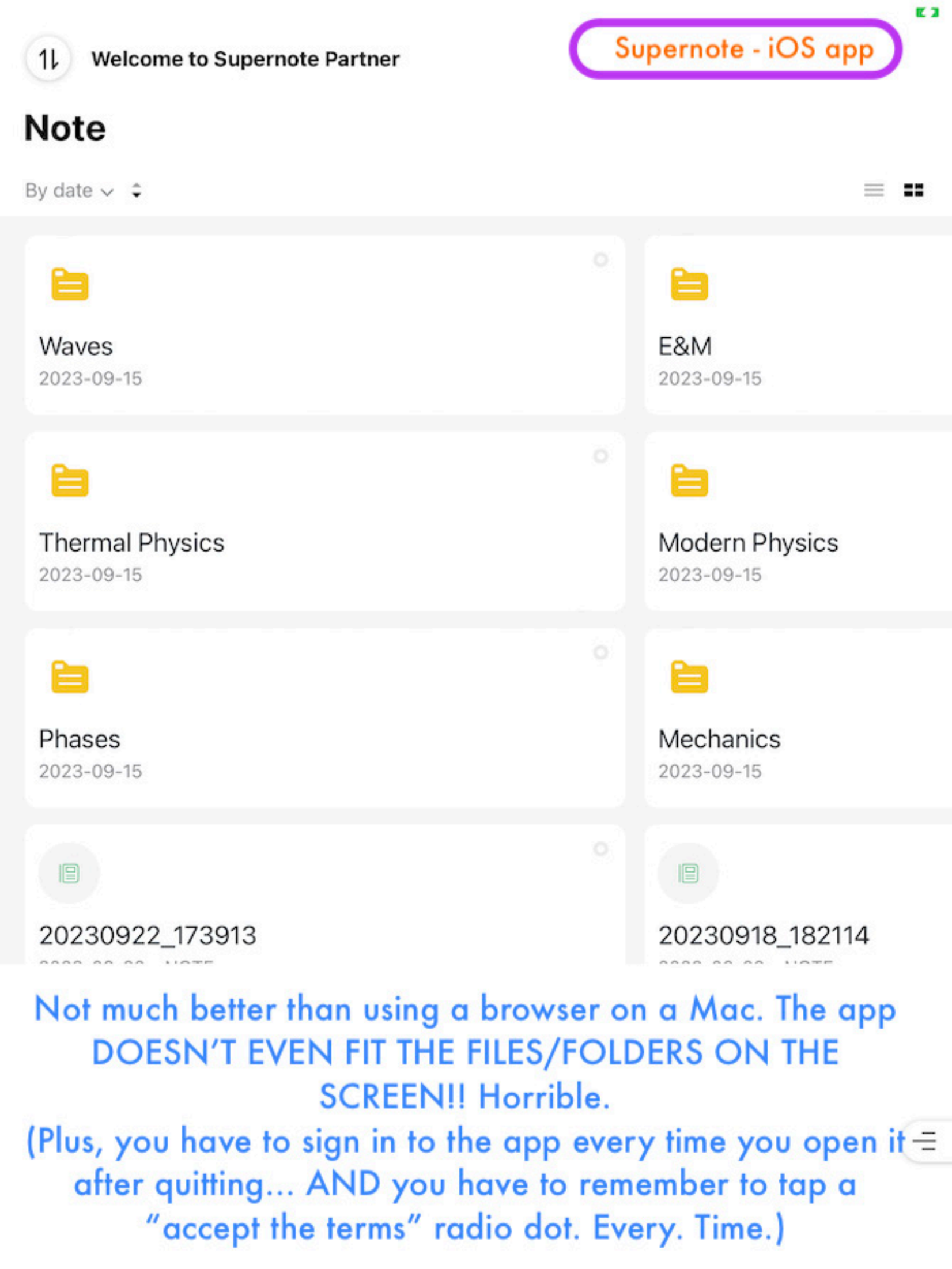


Recognition Test
2 pages



Recognition Test.txt
2 pages

Very similar to the Mac, just a different location for the menus. Still very nice, clean, and efficient.



Not much better than using a browser on a Mac. The app DOESN'T EVEN FIT THE FILES/FOLDERS ON THE SCREEN!! Horrible.

(Plus, you have to sign in to the app every time you open it after quitting... AND you have to remember to tap a "accept the terms" radio dot. Every. Time.)