

Cursor AI в iOS разработке. Приложение "Приливы" Tide Times

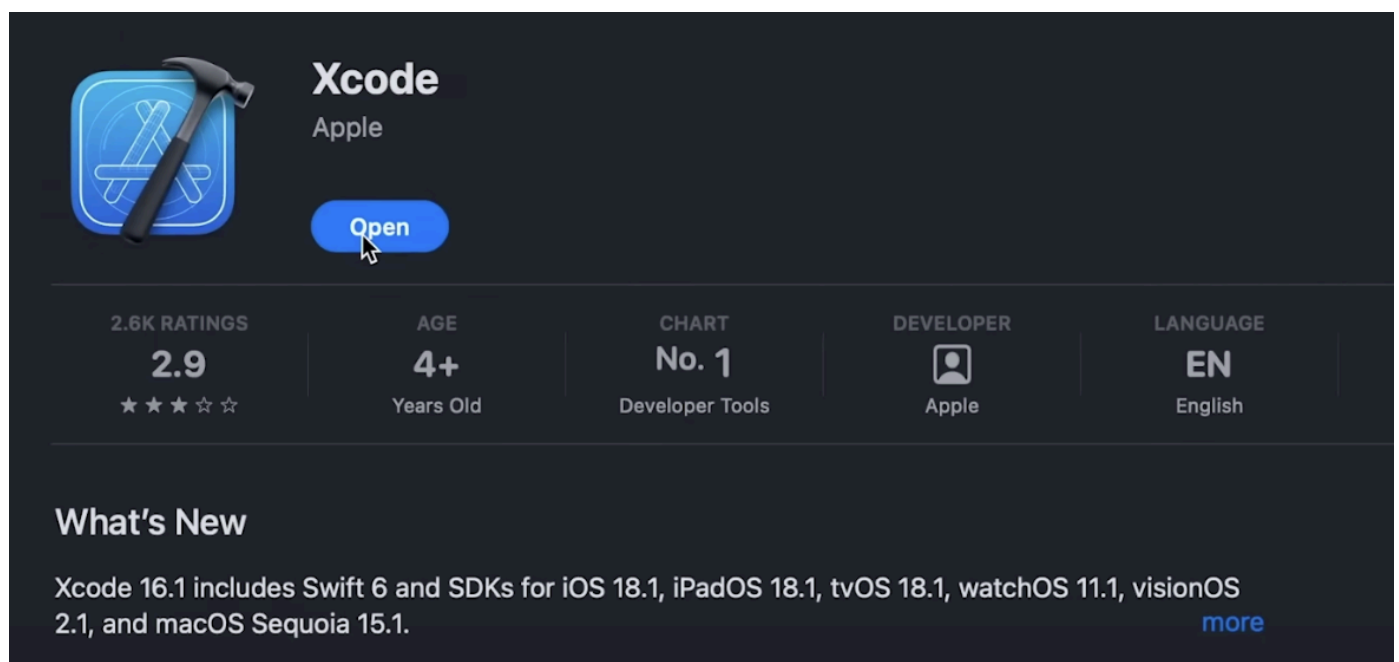
Основой для этой статьи стало видео ["I Made an iOS App in MINUTES with This AI Tool!"](#) ("Я сделал это iOS приложение в течении нескольких минут с помощью этого ИИ инструмента").

Я не верю, что без знания Swift и SwiftUI или UIKit можно создать полноценное iOS приложение, но мне было интересно, какие "фишки" использовал автор внутри Cursor AI, их там много: Chat, Composer, , Tab, Editor и Terminal.

Оказывается только Composer, который работает с целым проектом.

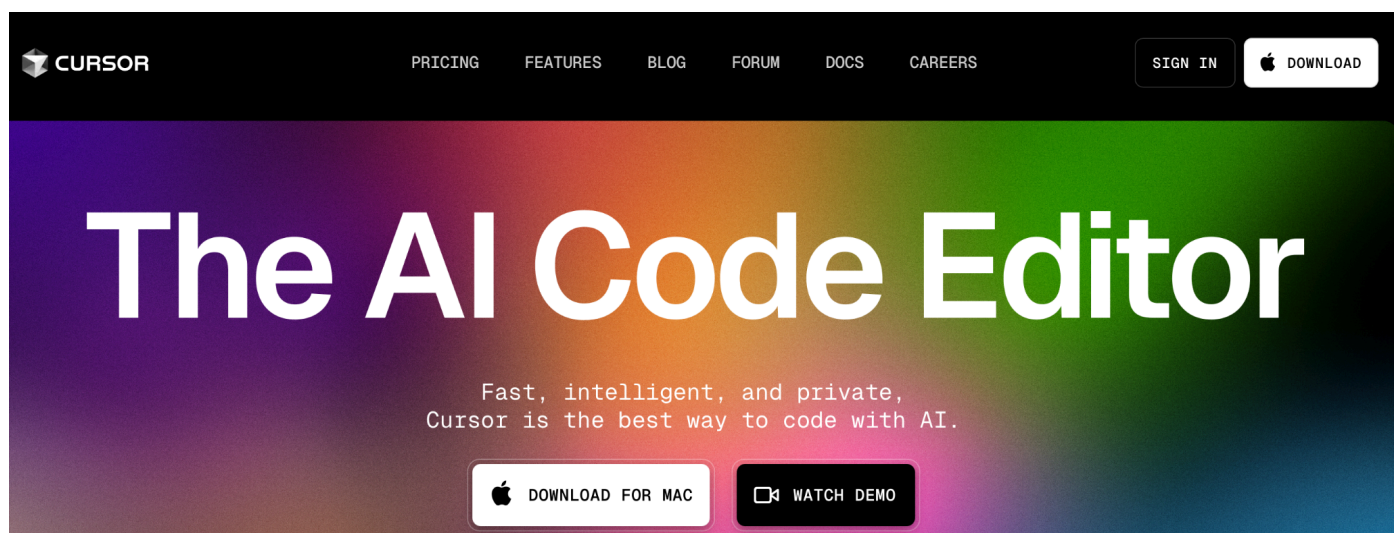
Вам нужен Mac - любой Mac Studio, MacBook, iMac.

Вам нужно будет загрузить Xcode из Apple App Store.

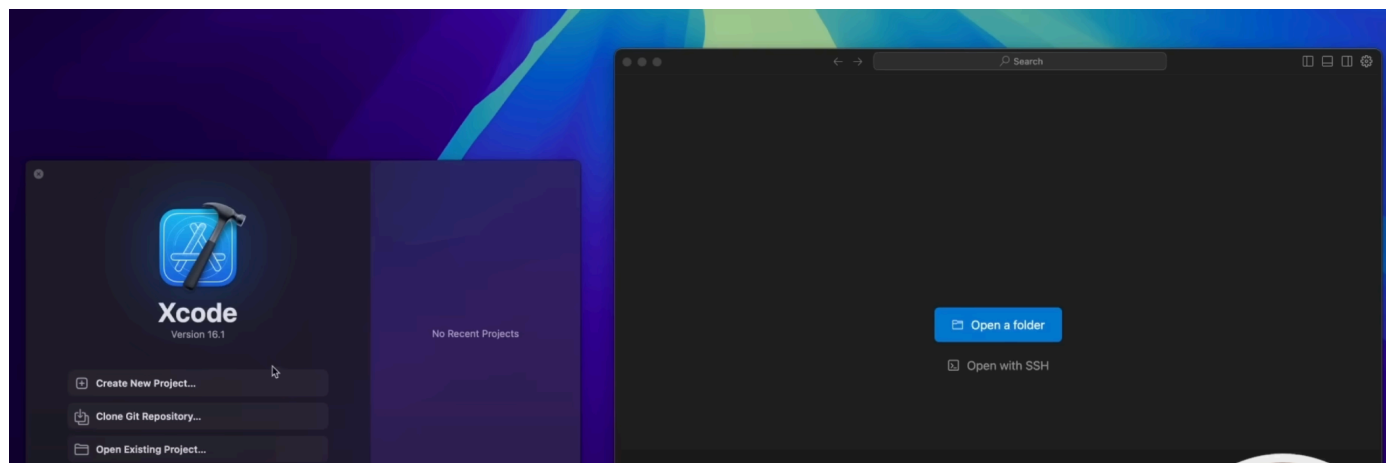


Это приложение Apple, и оно поможет вам хорошо писать код. Вы не будете писать код, ИИ сделает это за нас, но он вам понадобится для создания приложения.

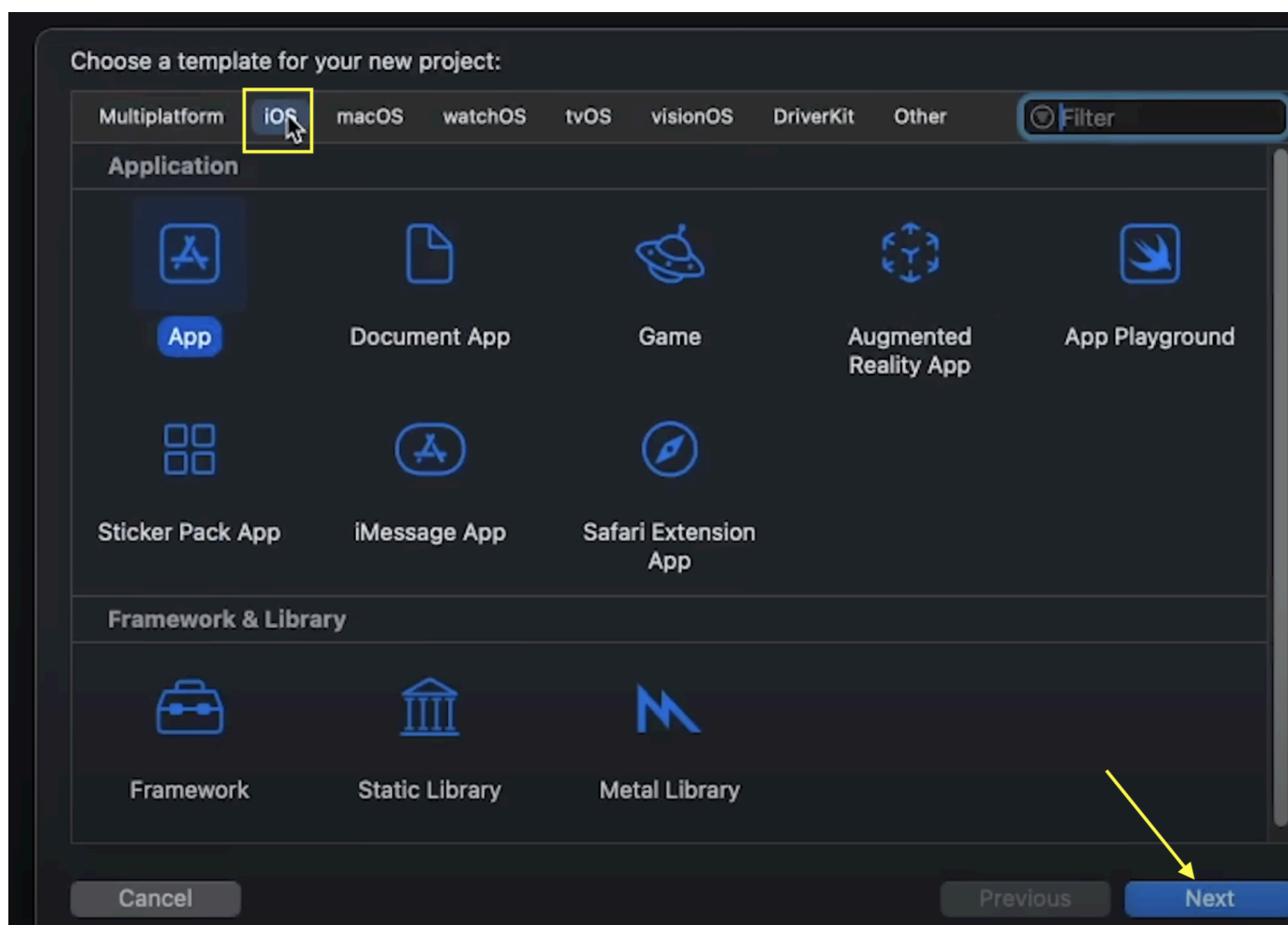
Загружайте [Cursor AI](#).



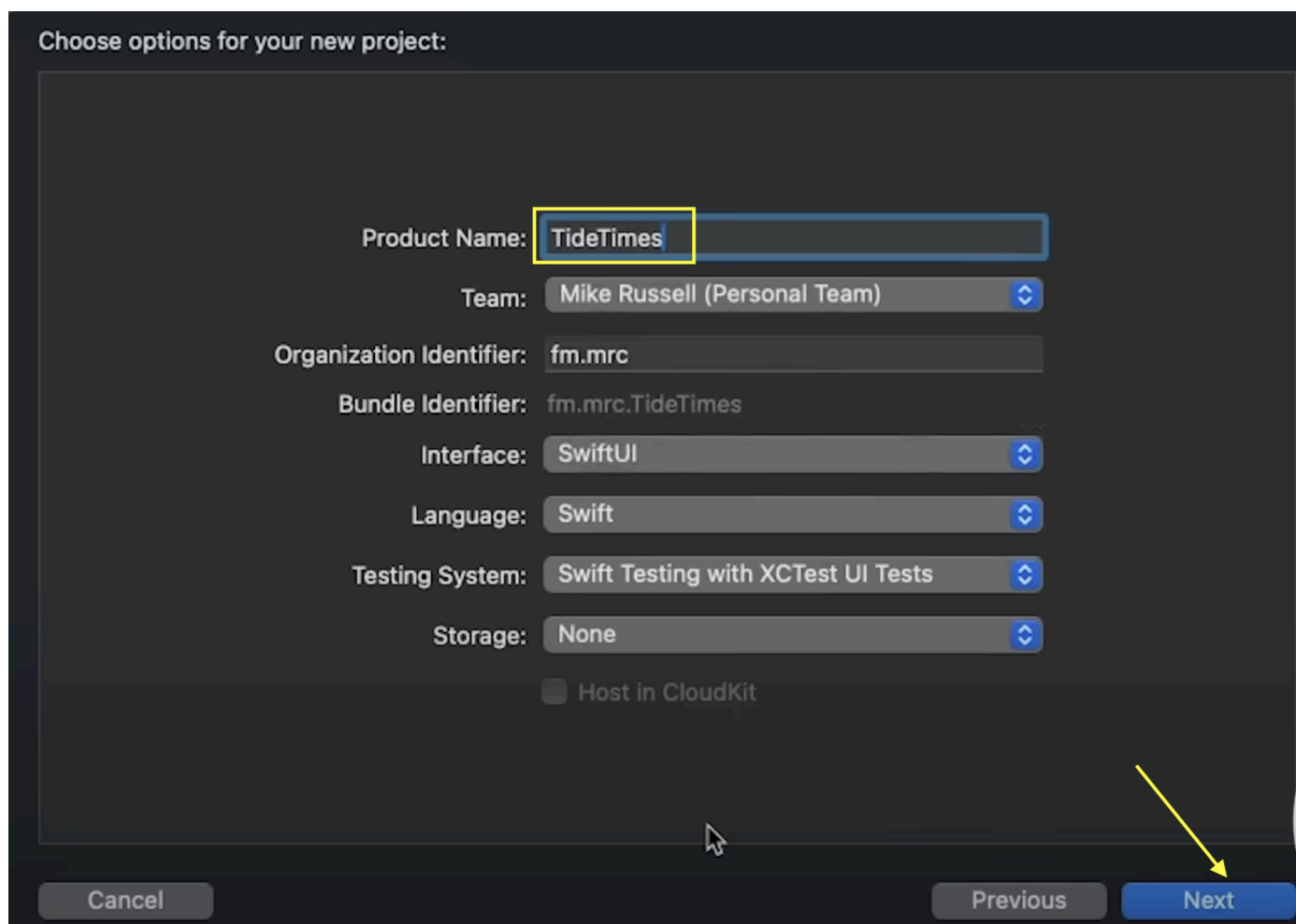
Все, что вам нужно сделать, это создать новый проект в Xcode:



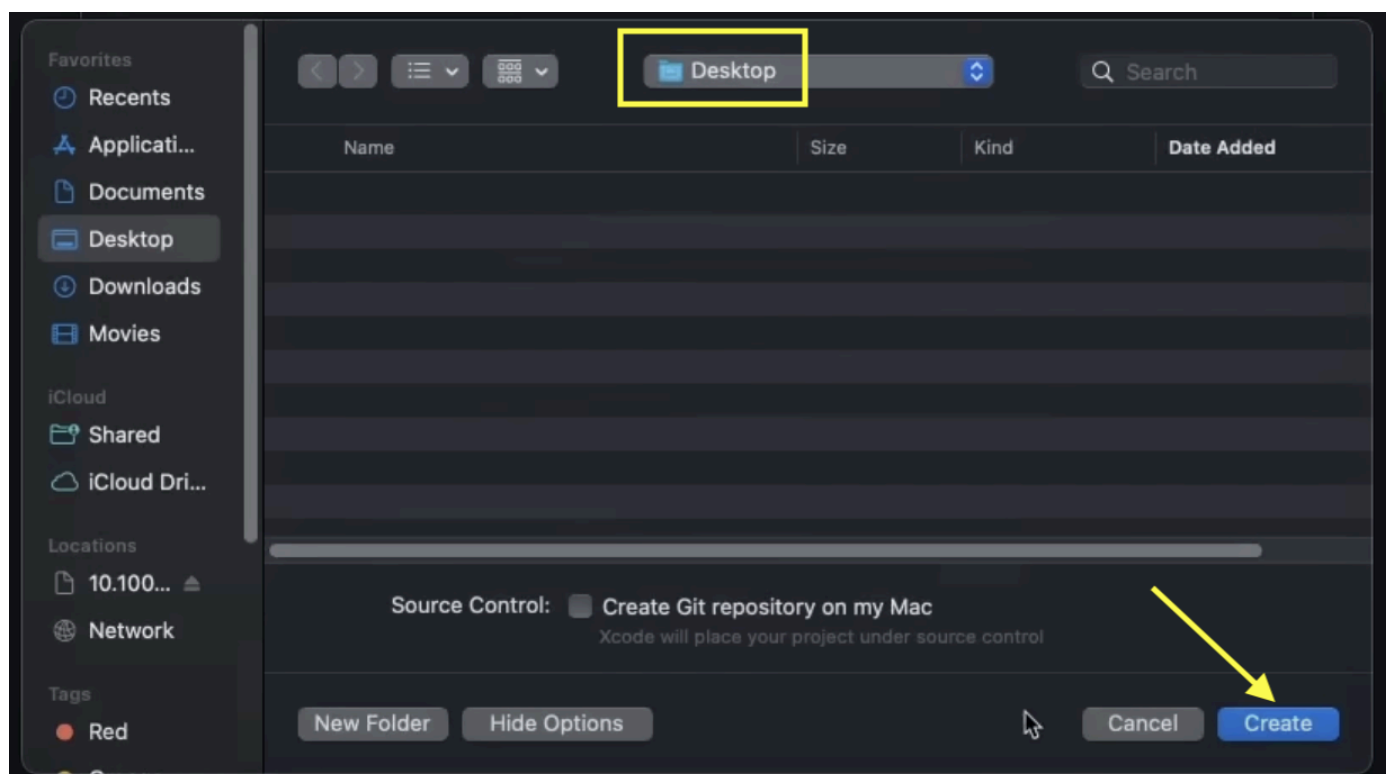
Убедитесь, что вы создаете iOS приложение:



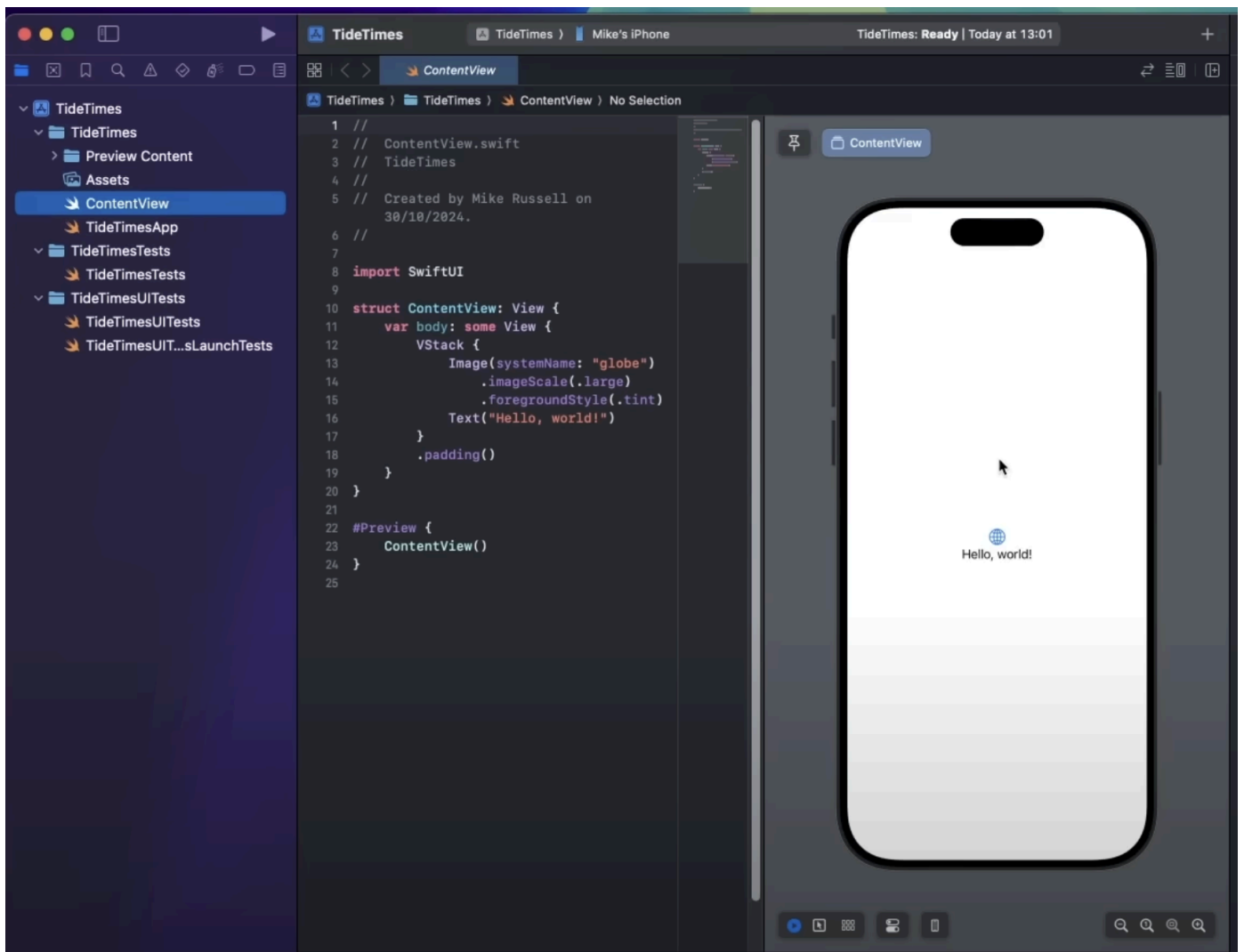
... и называем его Tide Times:



Просто нажмите **Next** (“Далее”), и затем вам будет предложено, где вы хотите сохранить файлы для приложения? Я выберу свой рабочий стол **desktop**, потому что это просто.



Я нажму **Create** (“Создать”), и вы увидите, что там создана папка **Tide Times**.

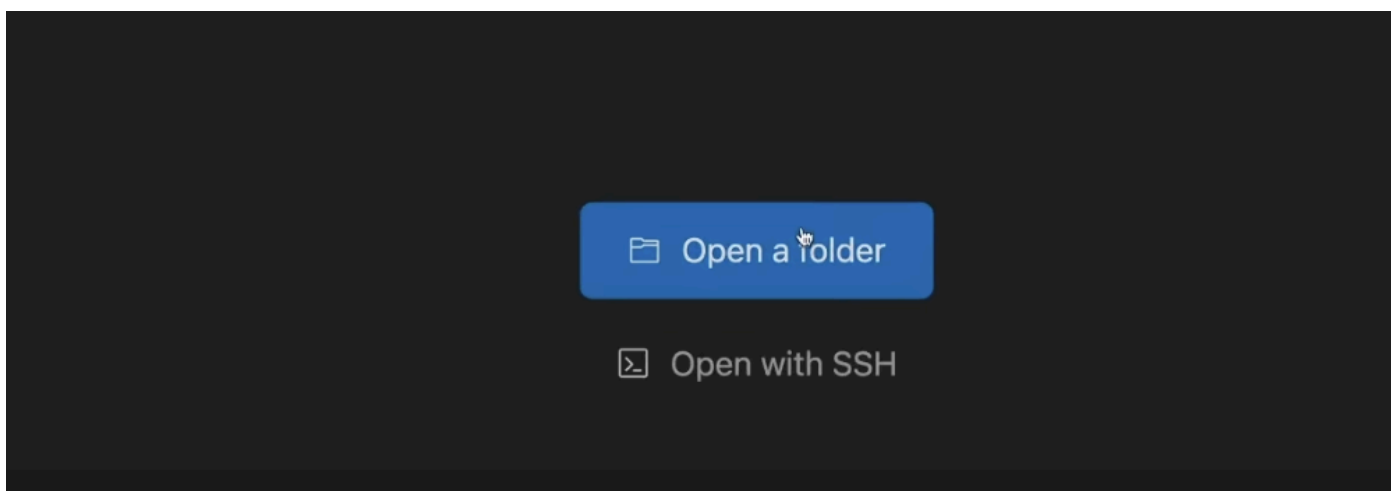


Xcode также создал все начальные файлы для моего приложения и вы видите **Hello World!** Да, верно, он дает вам пример кода, который отображает **Hello World!** на экране **iPhone**, что довольно круто.

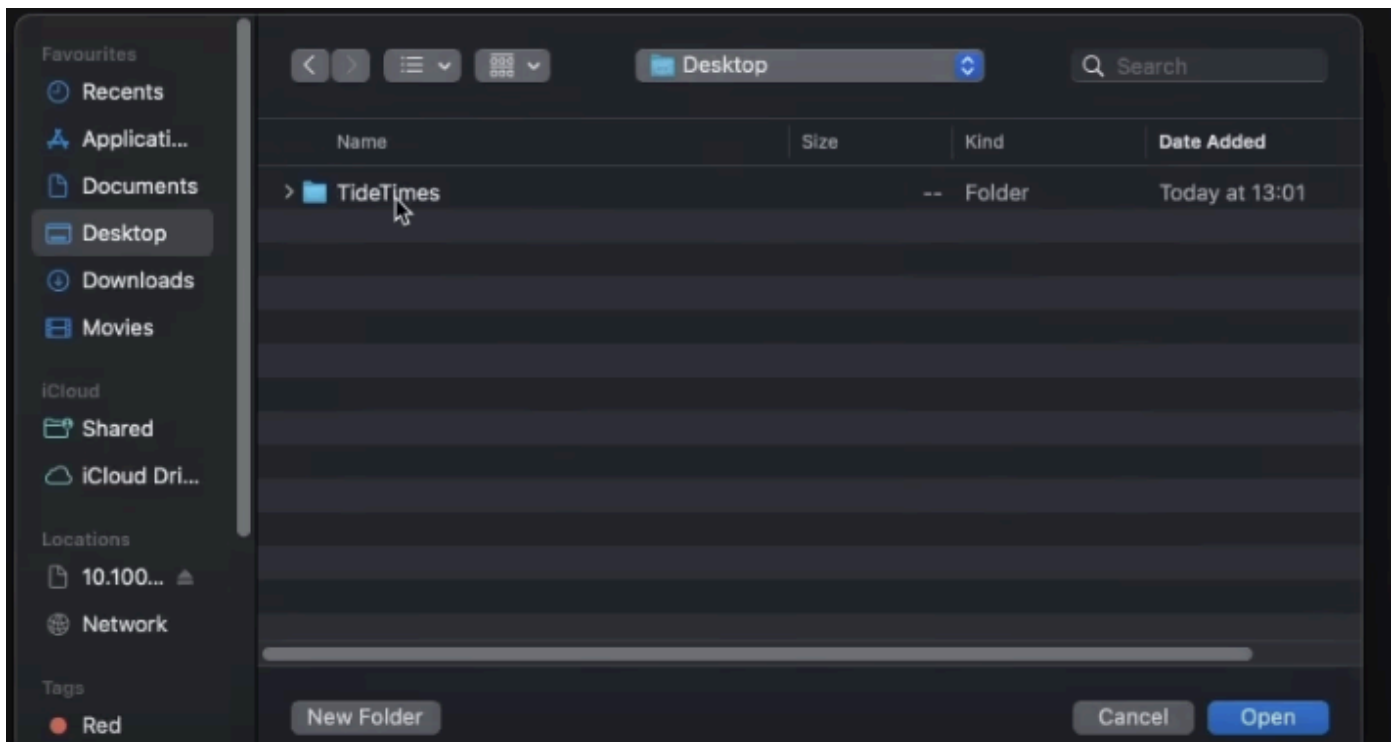
Итак, это показывает вам, как легко начать создавать iOS приложение, но мы не собираемся писать код. Мы собираемся использовать **Cursor AI**, чтобы сделать это.

Давайте посмотрим.

Запускаем **Cursor AI** и если это первый раз открываем папку с нашим Xcode проектом.

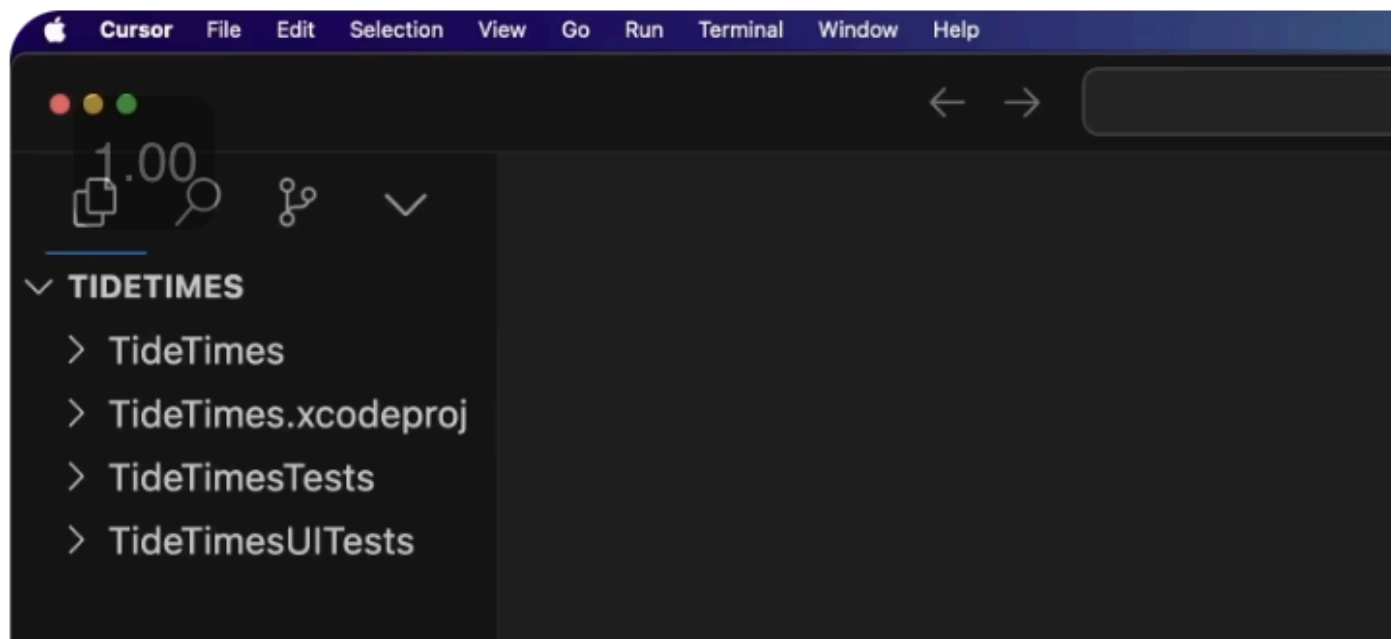


И, конечно, если я зайду на свой рабочий стол **desktop**, то там найду папку **Tide Times**.

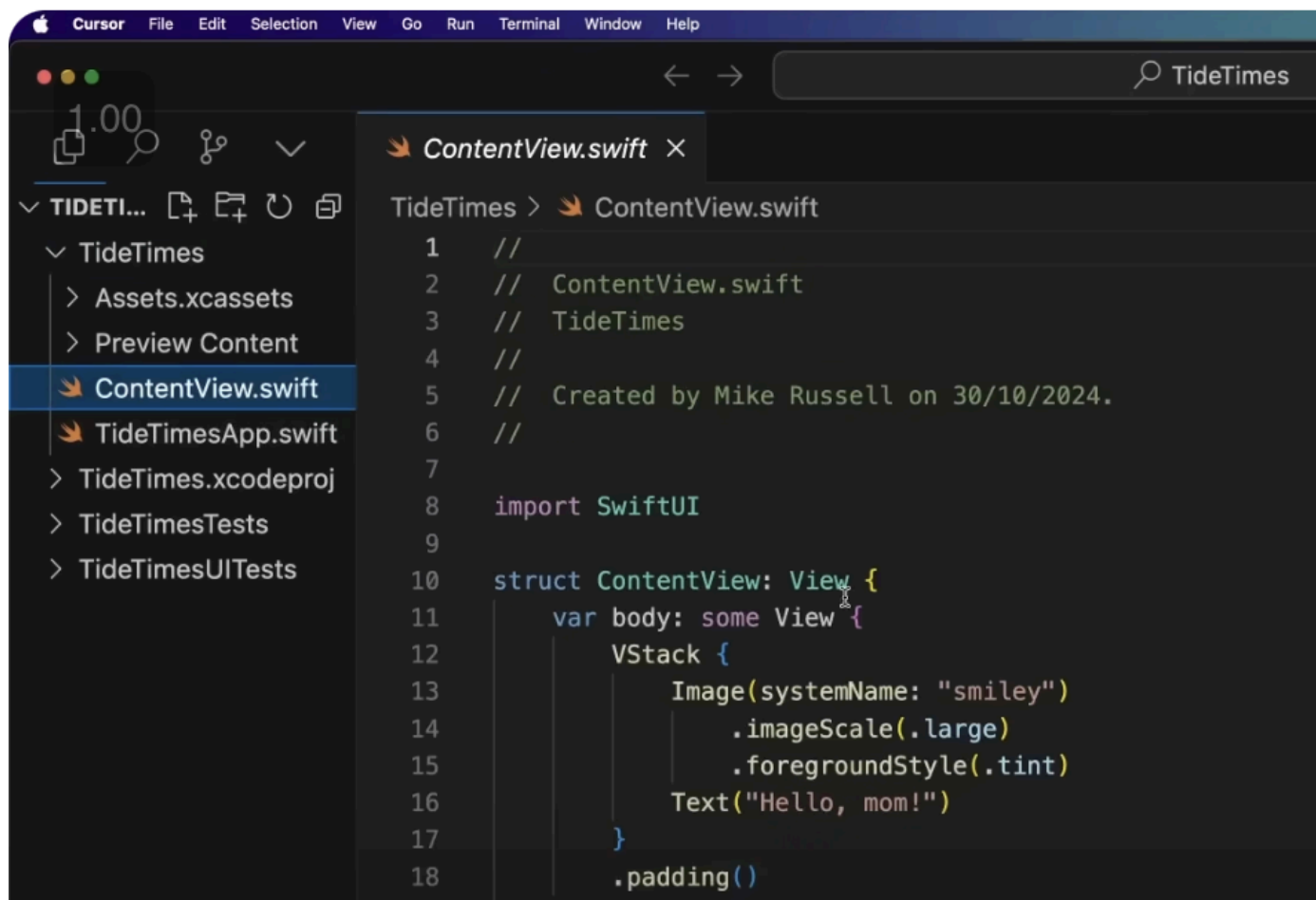


Это очень просто.

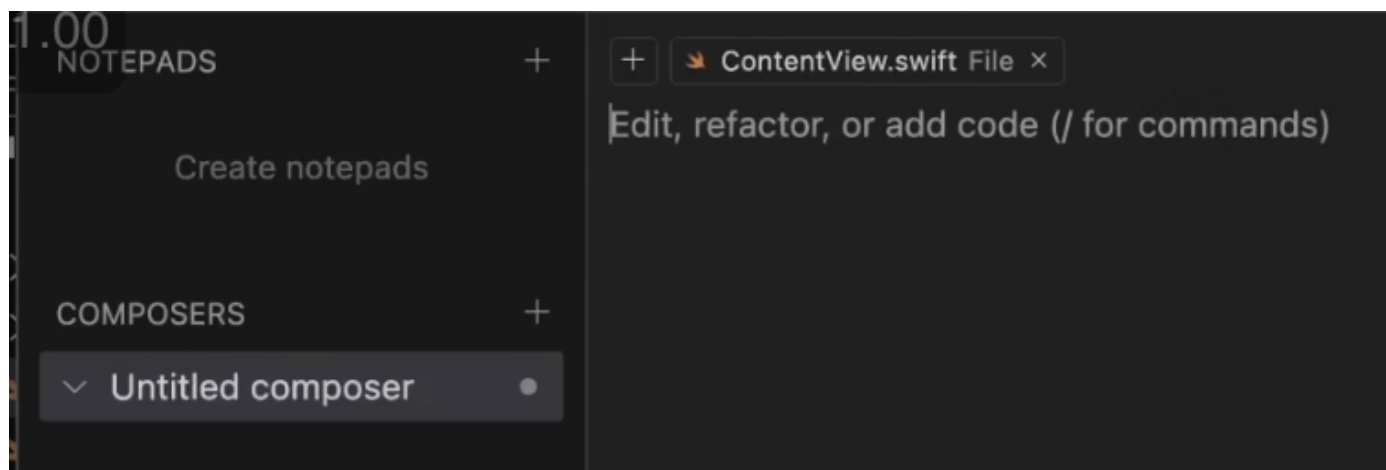
Нажимает на **Open** (“Открыть”) и все файлы подсоединились к Cursor AI.



Если я их открою, то это будут папки, а это файлы кода. И на самом деле, если я открою это, то вы увидите там все, что я делал в Xcode. Всё это теперь привязано к Cursor AI.



Начнем с того, что запустим Composer с помощью **Shift + Cmd + I**. Вы получите **Untitled Composer**.

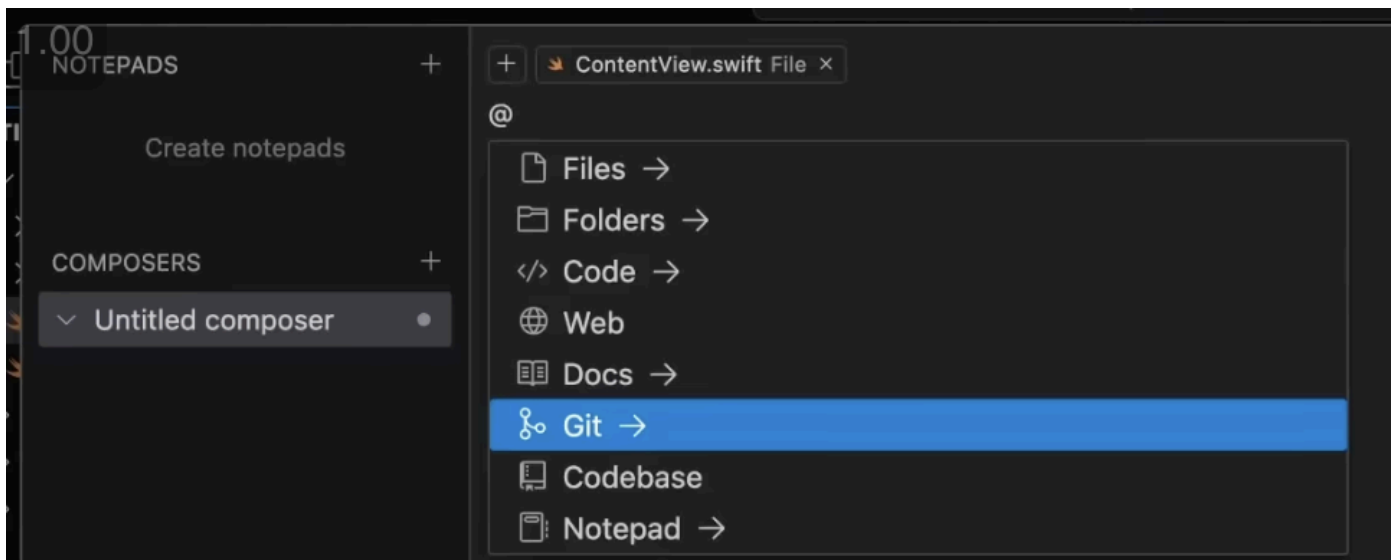


Вы можете использовать его, чтобы начать писать код.

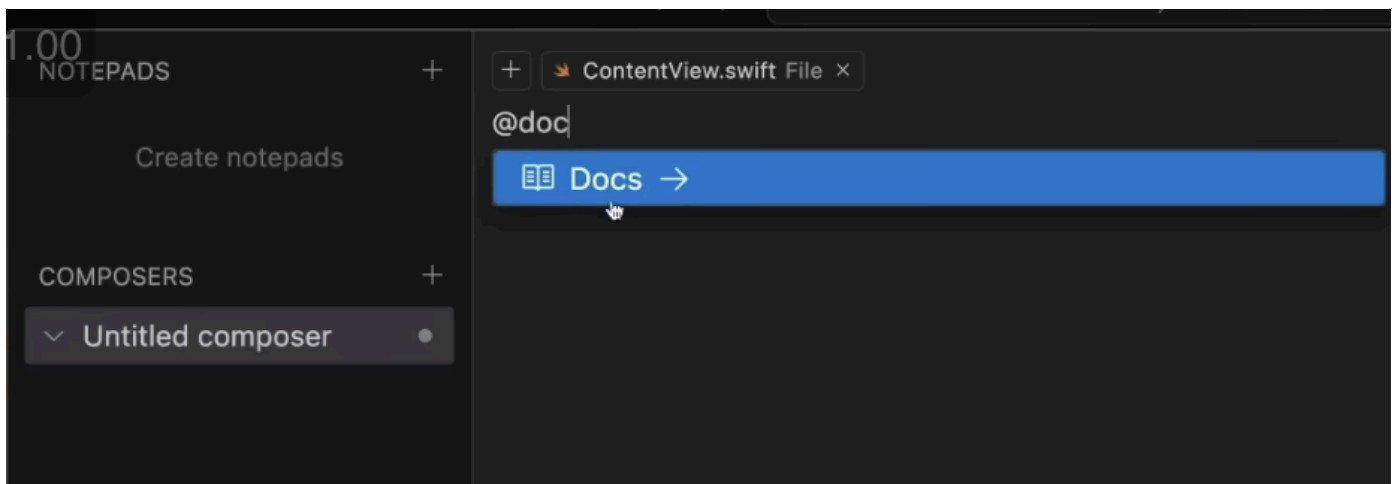
Ну, в первую очередь, я хочу убедиться, что следую рекомендациям Apple, лучшим практикам при создании моего приложения. Вы можете научить Cursor AI лучшим практикам Apple, чтобы ваше приложение стало идеальным.

Как вы это делаете?

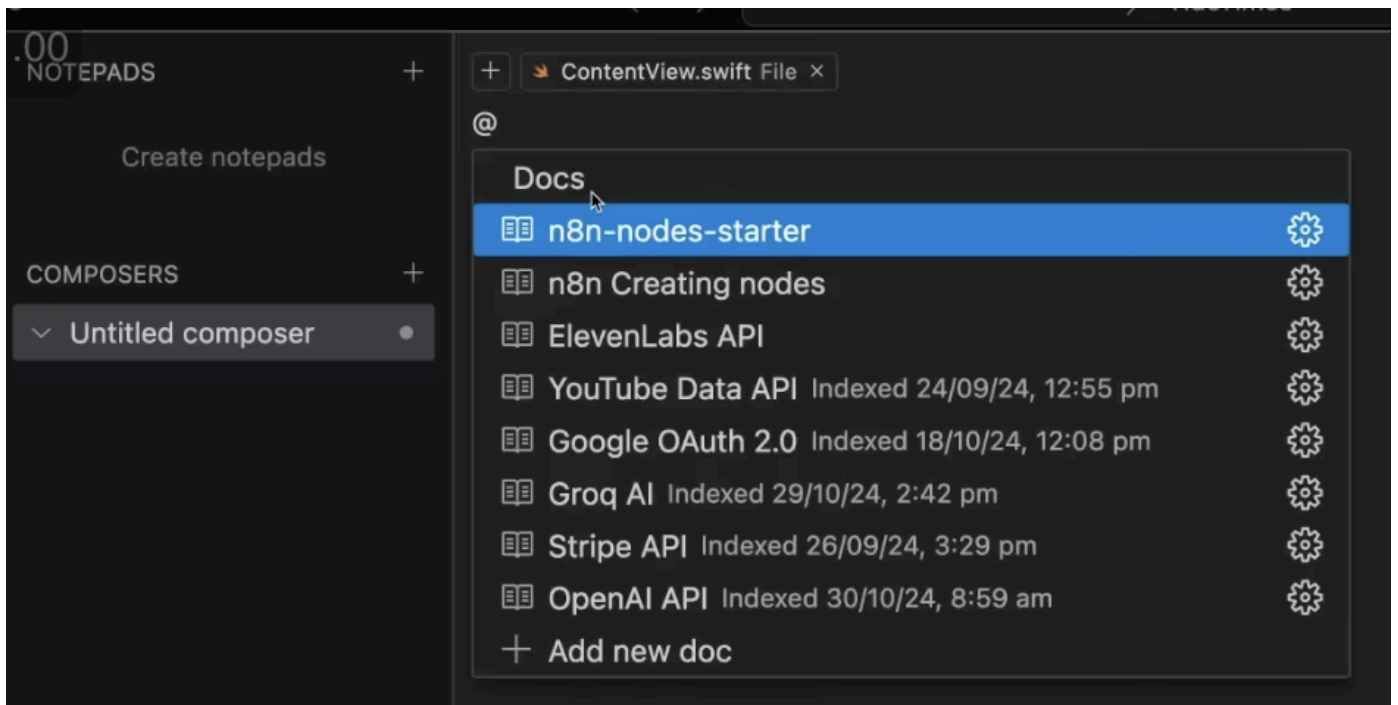
Введите символ **@** в Cursor Composer:



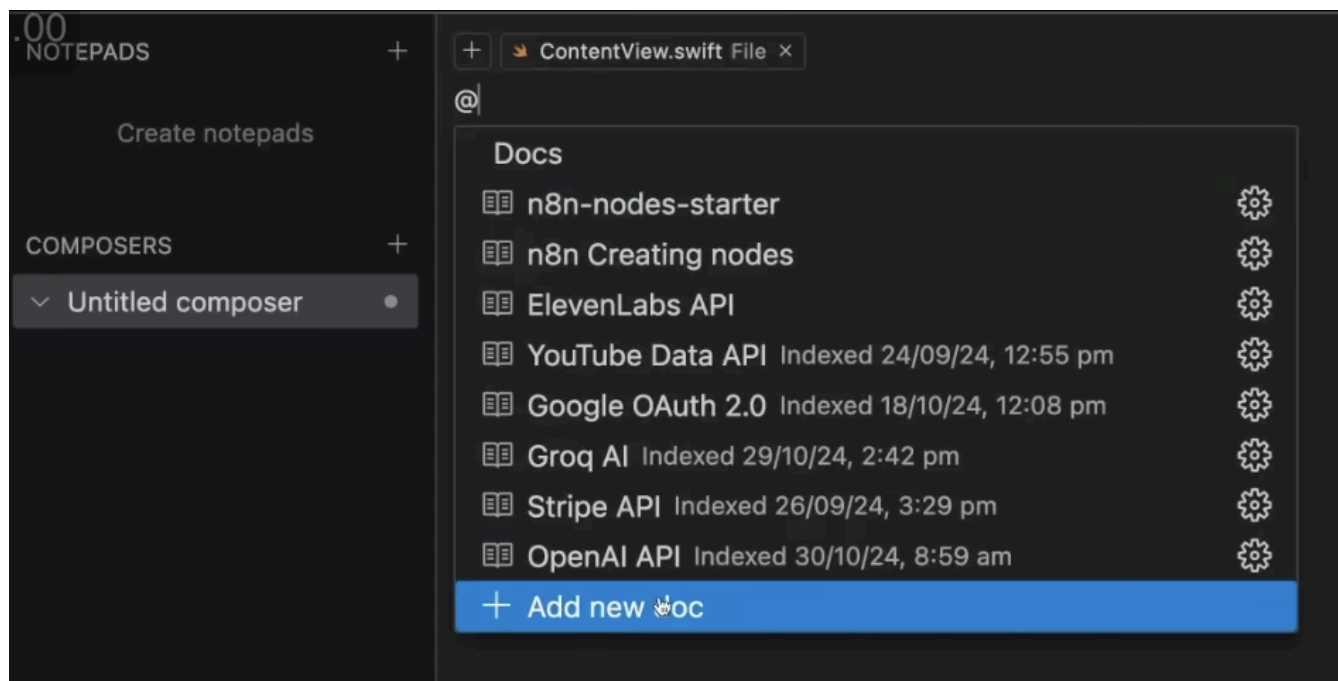
... и начните вводить @Docs:



и вы увидите @Docs выглядит так.

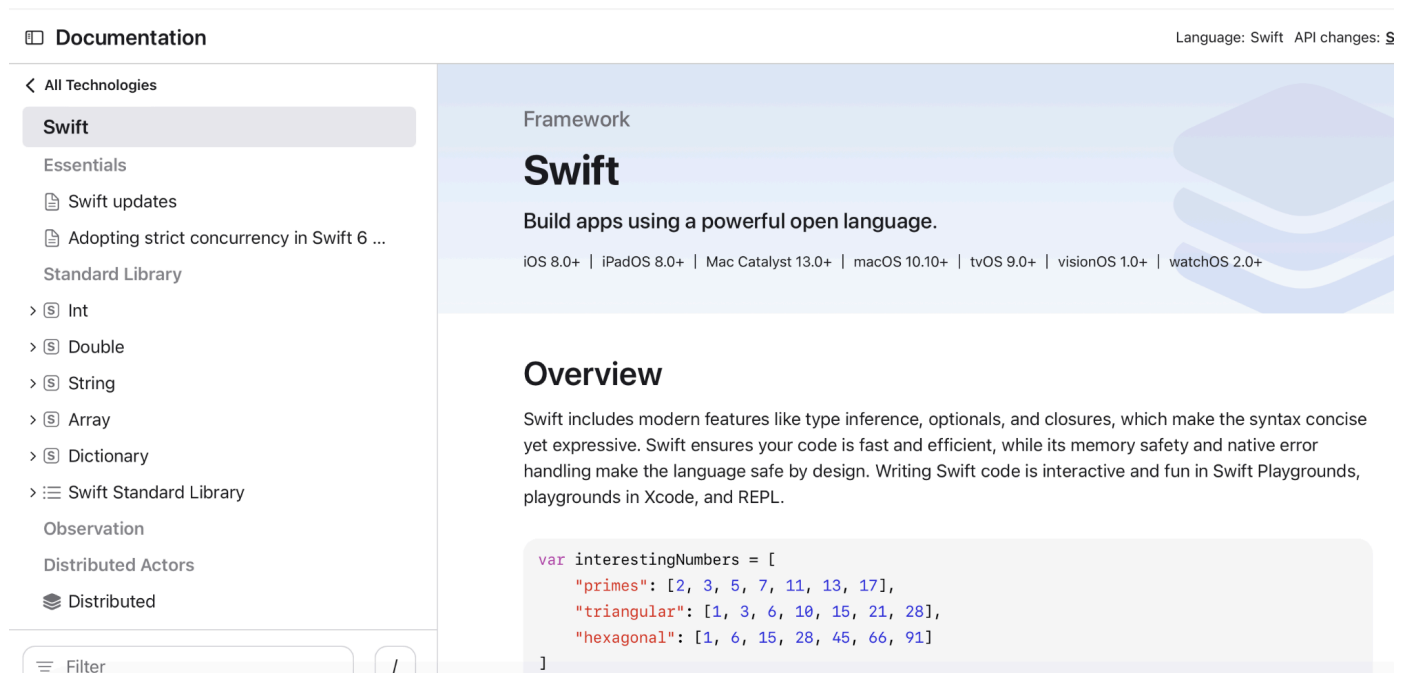


Теперь вы видите, что я уже обучал Cursor AI некоторым предыдущим документам, но я могу добавить новый документ здесь:

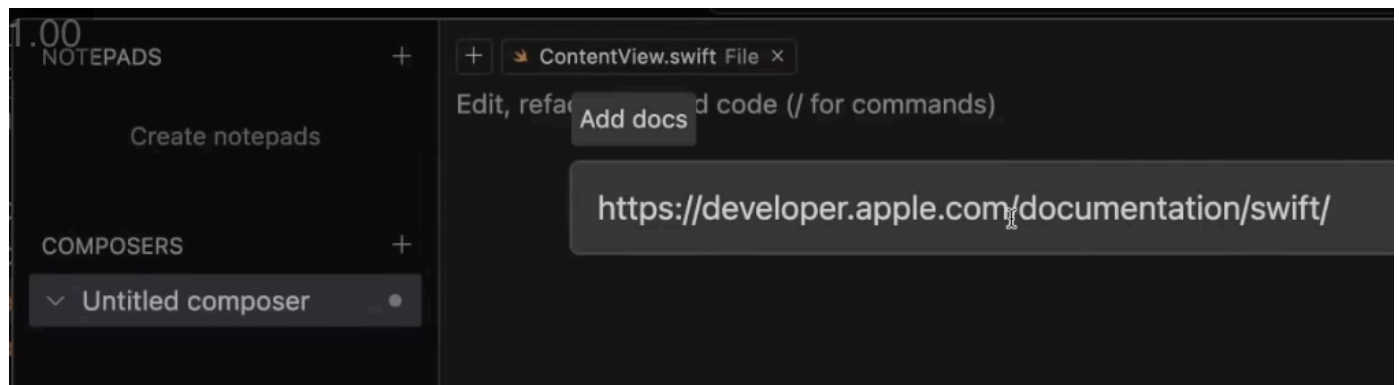


Я могу добавить веб-страницу. У меня на Chrome открыто несколько разных веб-сайтов.

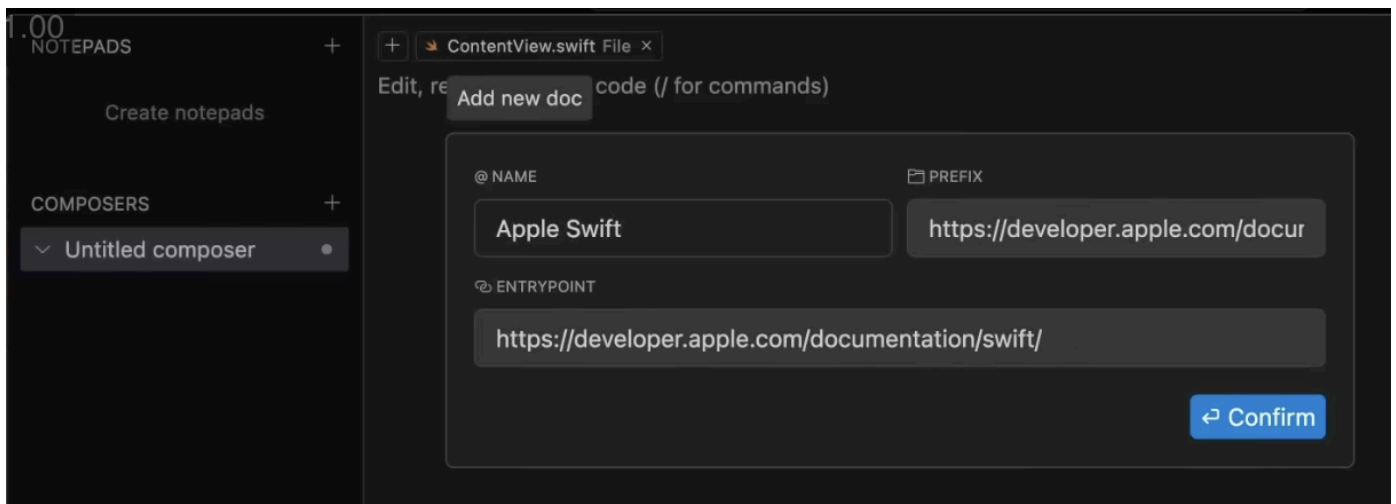
Во-первых, у меня есть документация Swift <https://developer.apple.com/documentation/swift/>



Итак, я скопирую URL-адрес Apple Swift, и вставлю его сюда:



Дадим ему имя **Apple Swift** и нажмем “Confirm” (“Подтвердить”):



И он начнет обучать Cursor AI Apple Swift.

Теперь я хочу обучить Cursor AI еще нескольким документам, включая “UI design Do's and Don'ts” (“Советы по дизайну пользовательского интерфейса”) от Apple

<https://developer.apple.com/design/tips/> :

Design

OverviewPathwayWhat's newGuidelinesResoi

UI Design Dos and Don'ts

Engaging user experiences are built on a foundation of solid interface design. Before you start coding, consider these fundamental design concepts for building clean, efficient interfaces for a broad set of users.

InteractivityReadabilityGraphicsClarity

Formatting Content

Create a layout that fits the screen of a device. Users should see primary content without zooming or scrolling horizontally.

[Learn more >](#)

✓

✗

Чтобы Cursor AI также следовал этим рекомендациям.

А это мы сохраним на потом, когда будем разрабатывать значок приложения

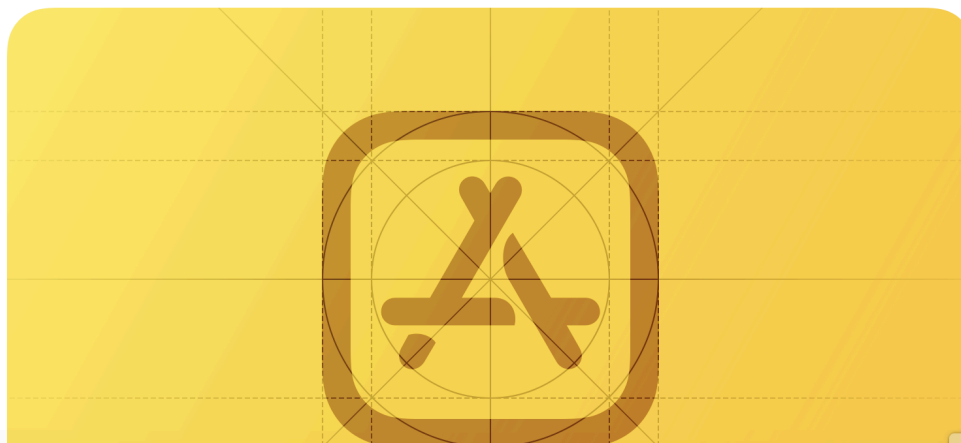
<https://developer.apple.com/design/human-interface-guidelines/app-icons/> :

Filter

- > Getting started
- > Foundations
- > Patterns
- > Components
- > Inputs
- > Technologies

App icons

A unique, memorable icon communicates the purpose and personality of your app or game and can help people recognize your product at a glance in the App Store and on their devices.

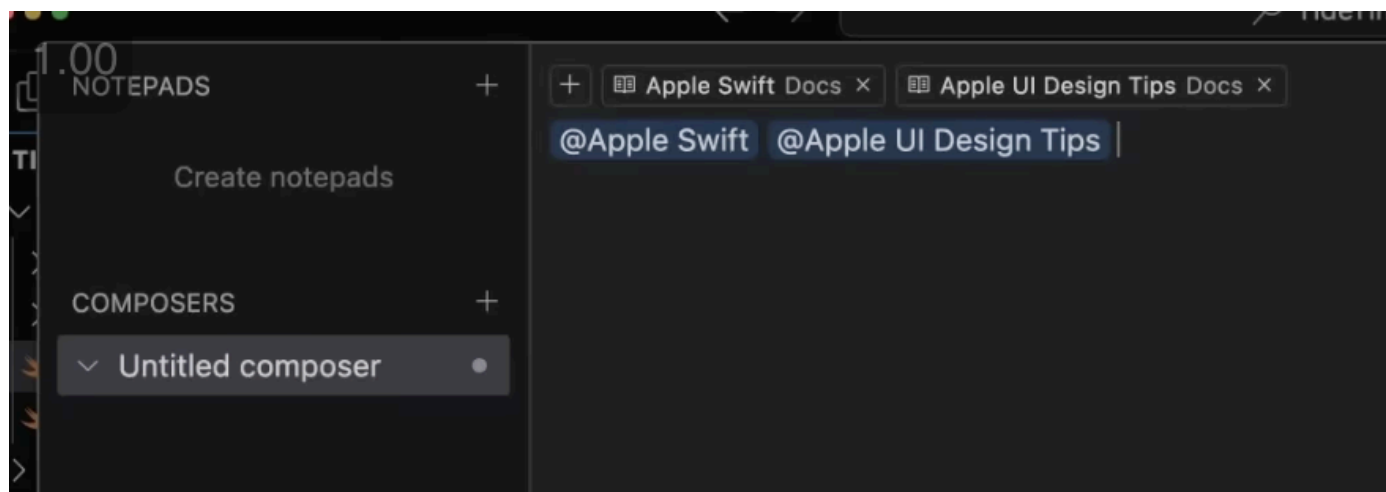


Мы сделаем это в конце видео.

До сих пор мы обучались в документации **@Apple Swift**.

Кроме того, “UI design Do's and Don'ts” (“Советы по дизайну пользовательского интерфейса Apple”):

@Apple UI Design Tips:

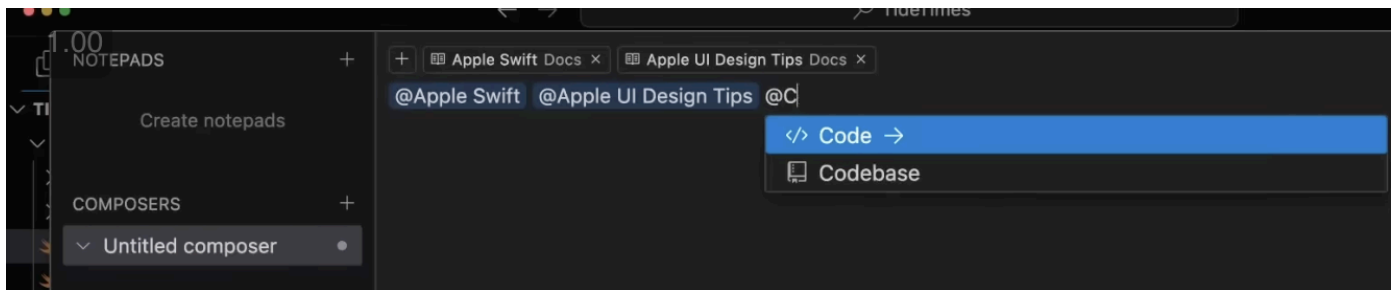


Cursor AI будет ссылаться на них при написании кода.

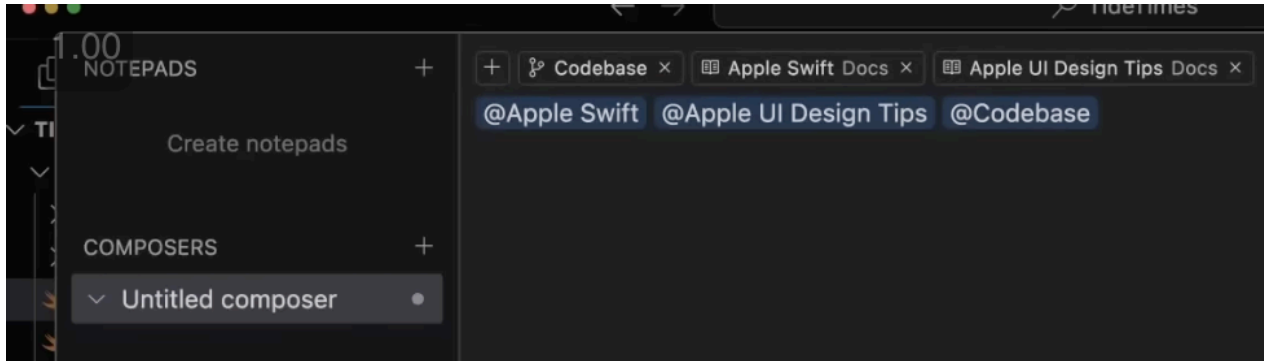
Итак, сегодня у нас все отлично. Мы не используем старые данные для обучения. Это последние новости на сайте Apple.

Еще один тег @ мы сделаем перед тем, как вставить описание нашего приложения.

И это @Codebase:

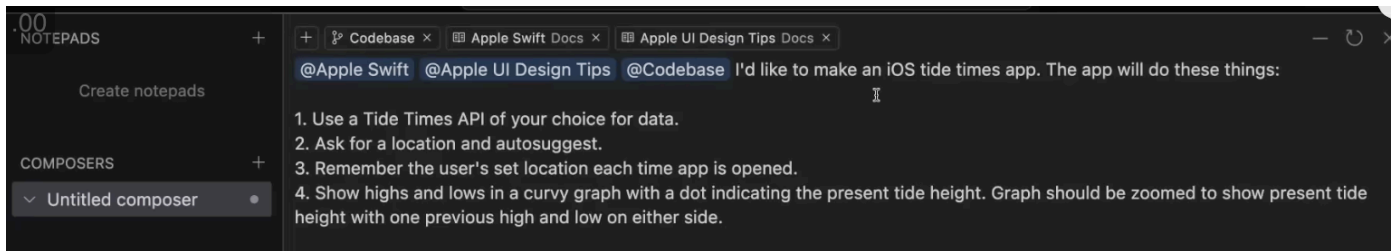


Кодовая база очень важна, потому что у нас есть существующие файлы, созданные Xcode, и мы хотим, чтобы Composer видел и редактировал их, создавая наше приложение:



Настройка описание приложения для Cursor AI

Теперь, когда все отмечено, я могу добавить описание приложения.



И я просто сказал, что хотел бы создать приложение Tide Times для iOS.

Оно будет делать следующее.

1. Использовать любое API Tide Times по вашему выбору для получения данных. По сути, это переход на веб-сайт с информацией о времени приливов и интеграция его в приложение. Вам не нужно беспокоиться о том, как это работает, потому что Cursor Compose сделает это за вас.
2. Затем он запросит у пользователя местоположение или предложит его.
3. В идеале он будет запоминать это местоположение каждый раз при повторном открытии приложения,
4. И будет показывать максимумы highs и минимумы lows приливов на криволинейном графике, на котором вы в виде пунктирной линии будете представлена текущая высота height прилива. График следует масштабировать, чтобы показать высоту прилива height с предыдущими максимум high и минимумом low,

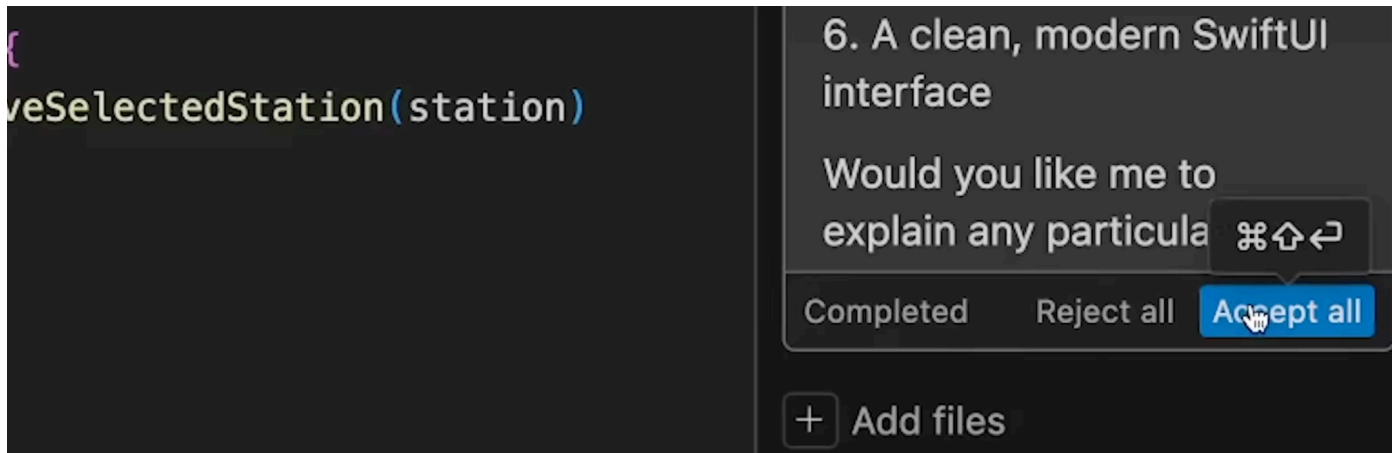
Это все, что мне нужно сделать, чтобы начать создавать свое приложение для iOS.

Я нажимаю Enter, и вы видите, что Cursor Composer приступает к работе.

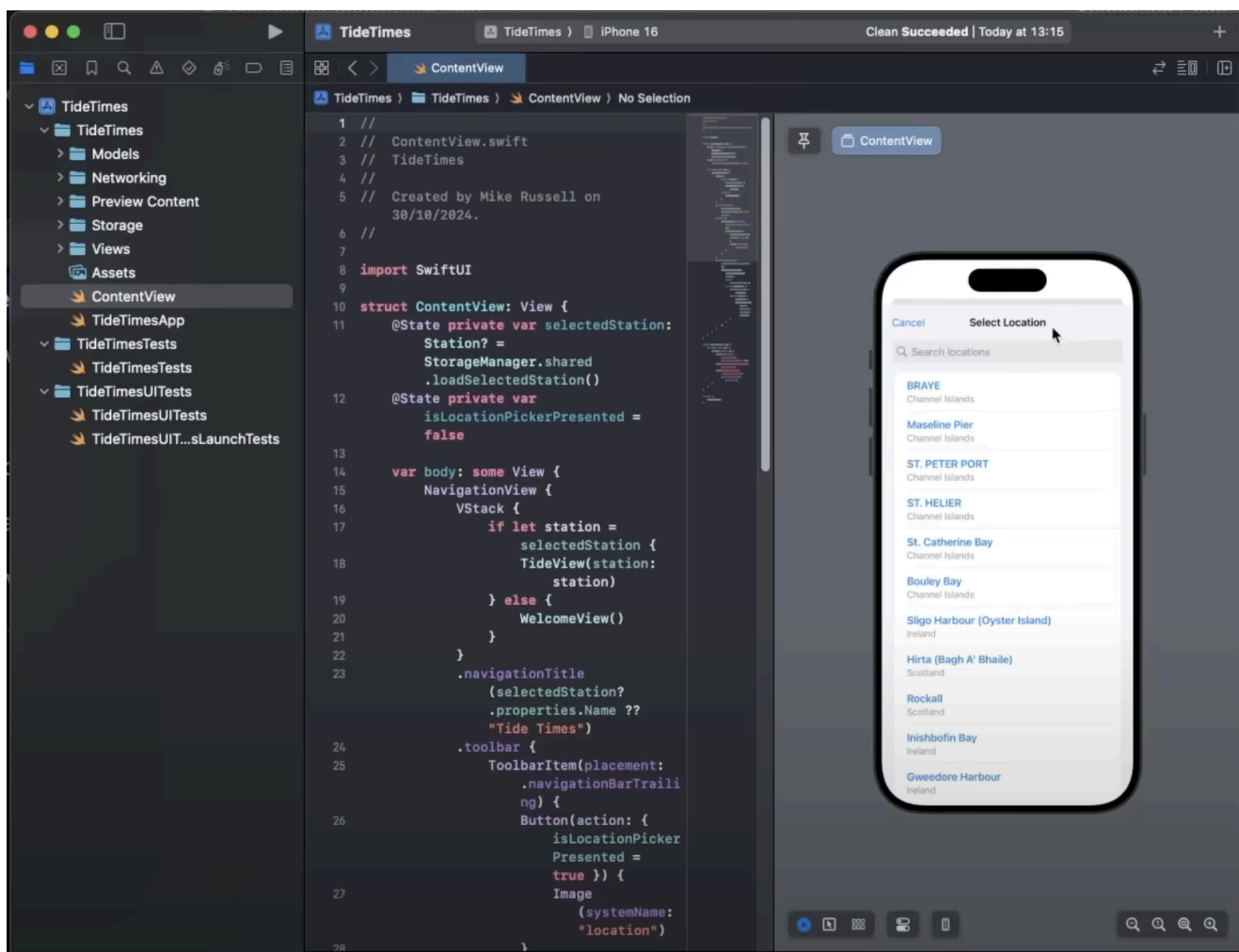
Как вы увидите, он пишет код прямо у меня на глазах.

Я пока не знаю, что делает этот код.

Он просто создает мое приложение в соответствии с лучшими принципами дизайна Apple.
Давайте нажмем Accept All.



В Xcode мы видим, что предварительный просмотр Preview обновлен:



И на самом деле он дает мне возможность выбрать местоположение.

Итак, давайте начнем вводить текст.

О, смотрите, вот Dover (Дувр). Мы можем нажать на Dover, хорошо.



Теперь у нас белый экран.

Так что в настоящее время данные не отображаются.

Мы вернемся к Cursor и снова запустим его, чтобы убедиться, что данные отображаются.

```

WelcomeView()
{
}
.navigationTitle(selectedStation?.properties.Name ?? "Tide Times")
.toolbar {
    ToolbarItem(placement: .navigationBarTrailing) {
        Button(action: { isLocationPickerPresented = true }) {
            Image(systemName: "location")
        }
    }
}
.sheet(isPresented: $isLocationPickerPresented) {
    LocationPickerView(selectedStation: Binding(
        get: { selectedStation },
        set: { newStation in
            selectedStation = newStation
            if let station = newStation {
                StorageManager.shared.saveSelectedStation(station)
            }
        }
    ))
}
}

```

4. Persistent storage of the selected location

5. Error handling and loading states

6. A clean, modern SwiftUI interface

Would you like me to explain any particular part in more detail or make any adjustments to the implementation?

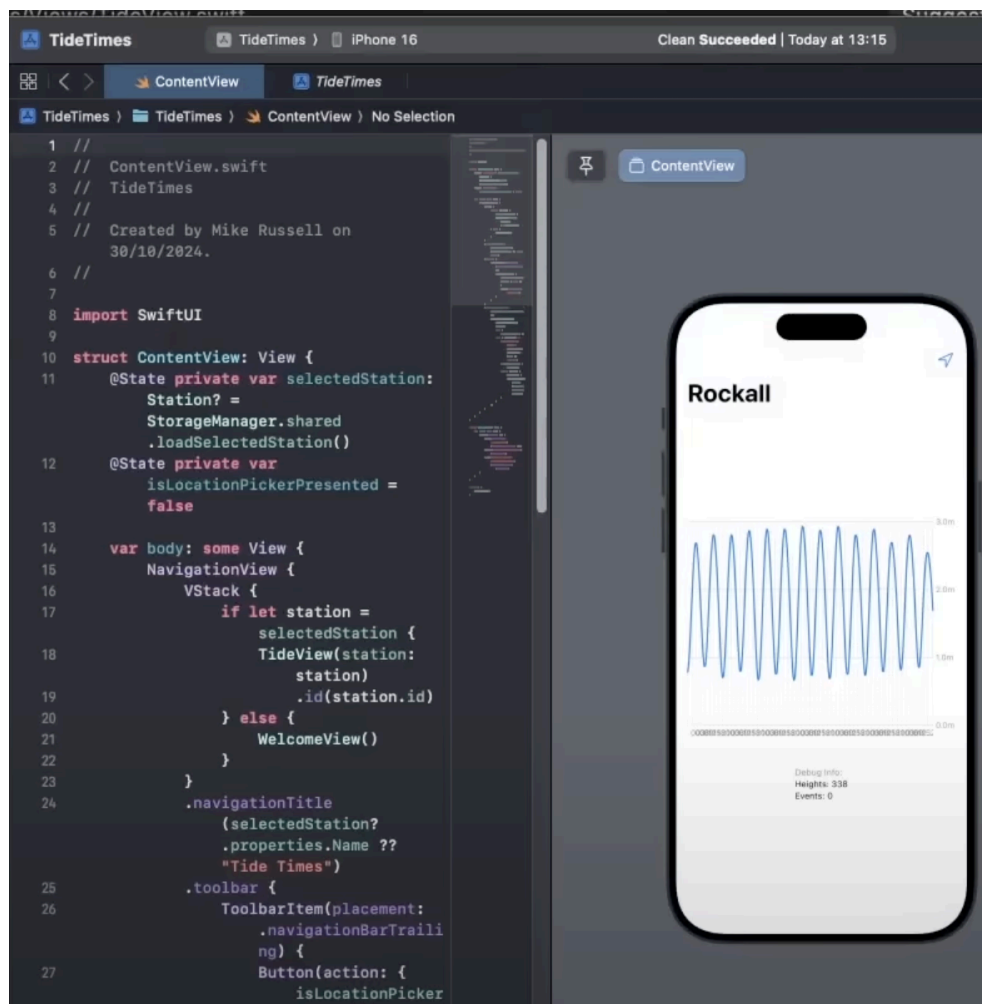
@Codebase no data shows after picking a location. Make this work.

Generating.. Cancel chat

Итак, я сказал @Codebase, что после выбора местоположения данные не отображаются, заставь это работать. Он собирается продолжить работу и обновить код, чтобы начать показывать информацию о приливах.

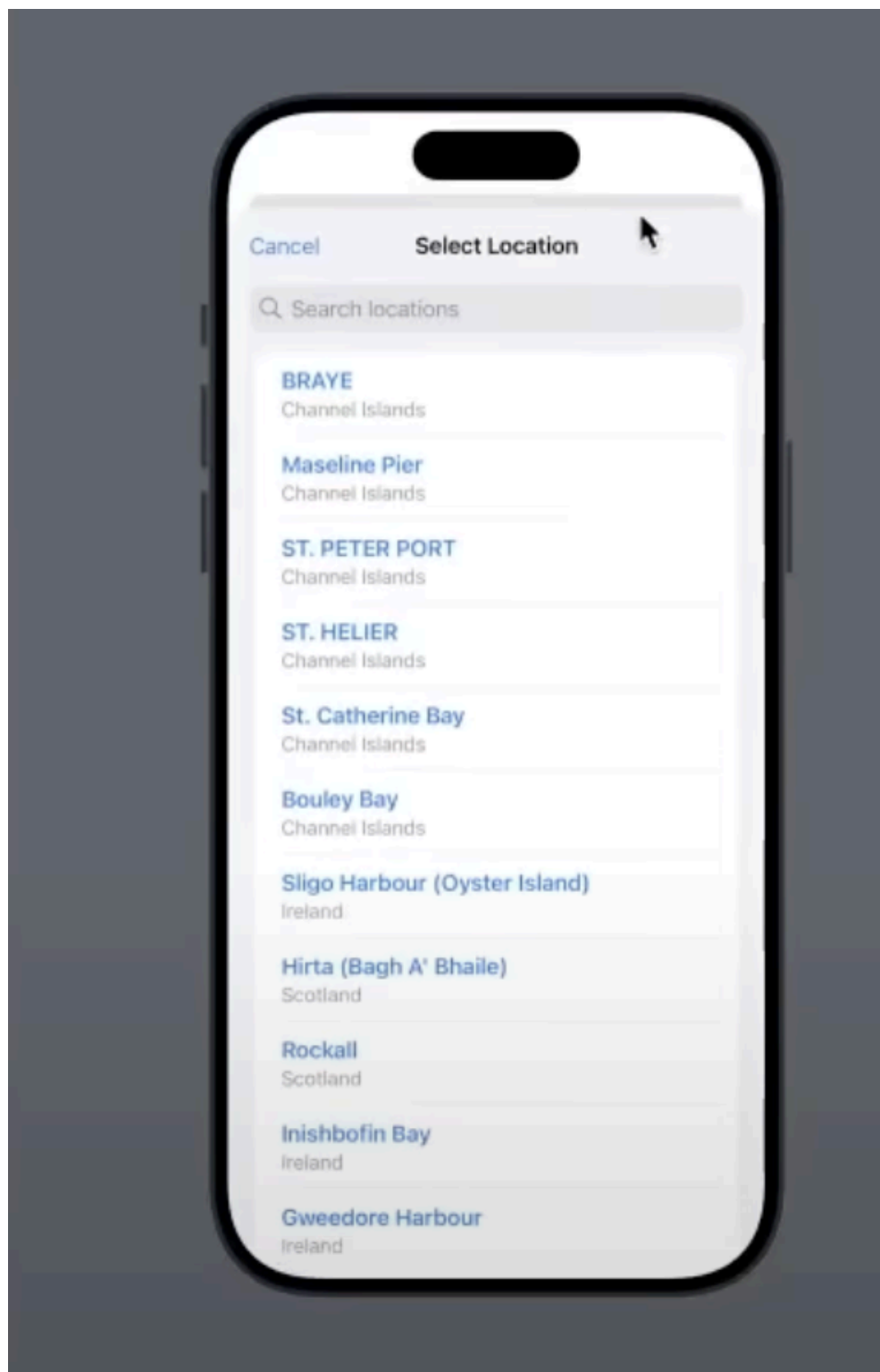
Хорошо, вернемся в Xcode.

Проект обновляется, и теперь действительно показывает область вместе с довольно волнистой таблицей времени приливов.

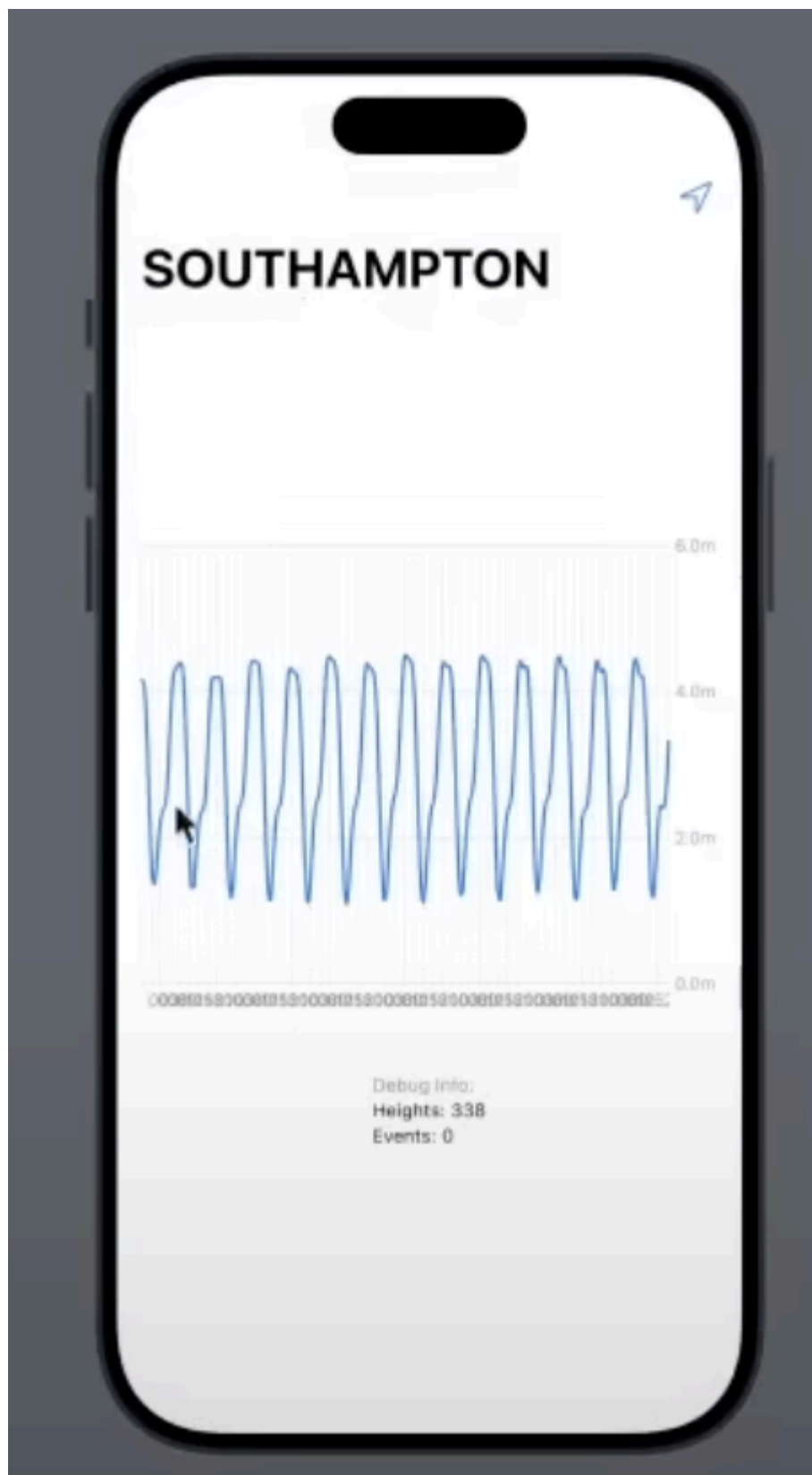


Даже есть некоторая отладочная информация там внизу.

И я думаю, я могу выбрать значок местоположения в правом верхнем углу и ввести другую область:



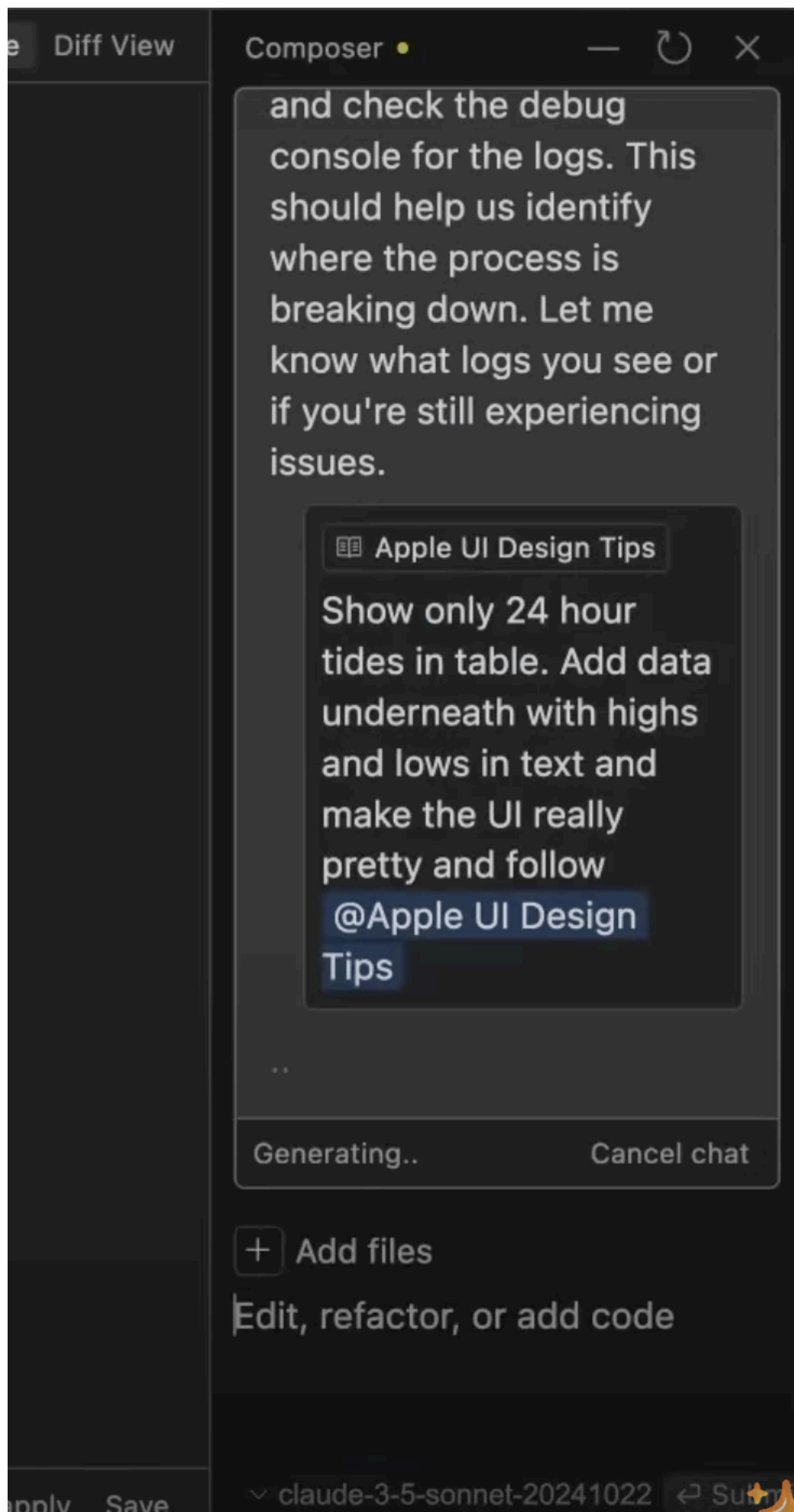
Итак, давайте введем Southampton (Саутгемптон), и БУМ!!, у нас есть время приливов для Саутгемптона:



Дальнейшая доработка дизайна пользовательского интерфейса с помощью искусственного интеллекта Cursor AI

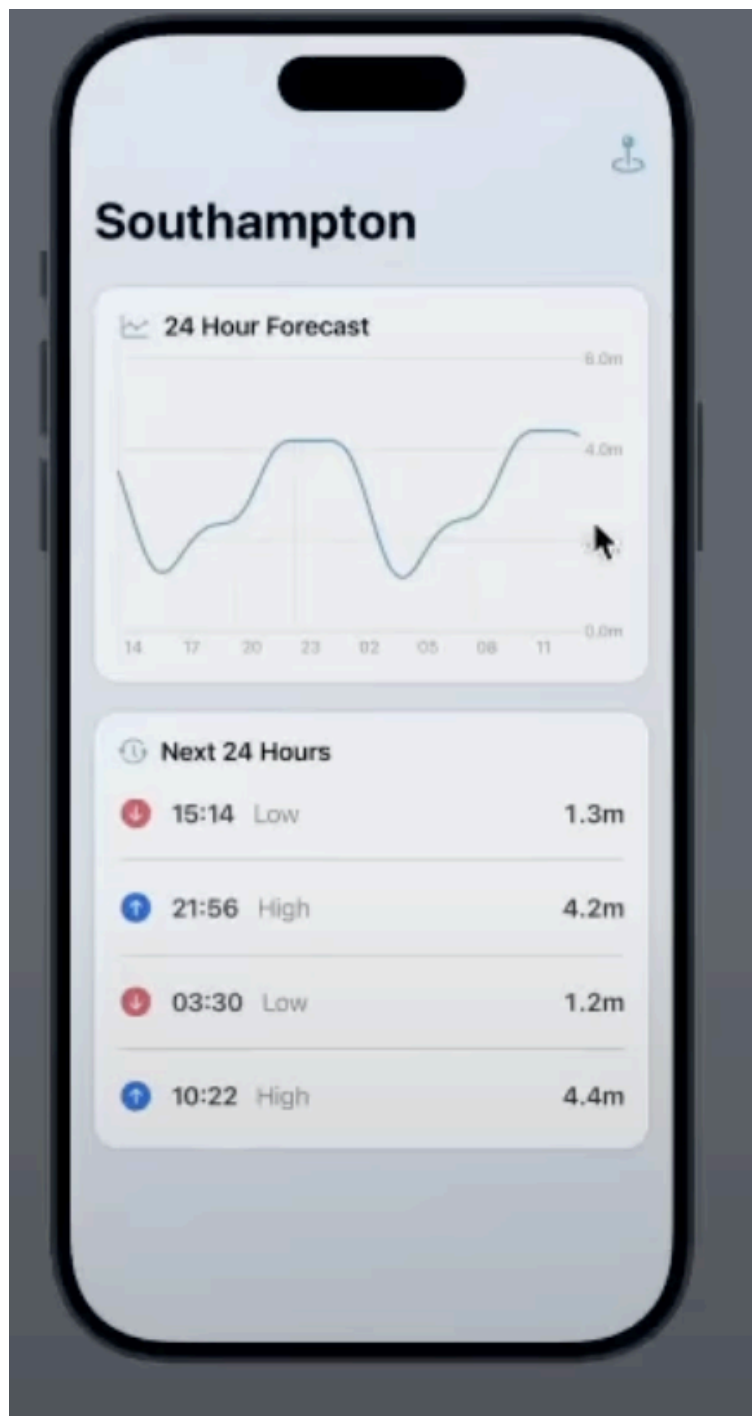
.Это не совсем соответствует лучшим практикам Apple в дизайне, который выглядит уродливо. Теперь давайте исправим это в Cursor AI, чтобы в таблице отображались только 24-часовые приливы.

Добавим снизу данные с максимумами highs и минимумами lows в тексте и сделаем UI действительно красивым и будем следовать **@Apple UI Design Tips**:



Мы позволим Cursor AI работать и исправим все это и сделаем приложение намного красивее. Ладно, просто жду, когда эти изменения обновятся в моем приложении.

И посмотрите на это:

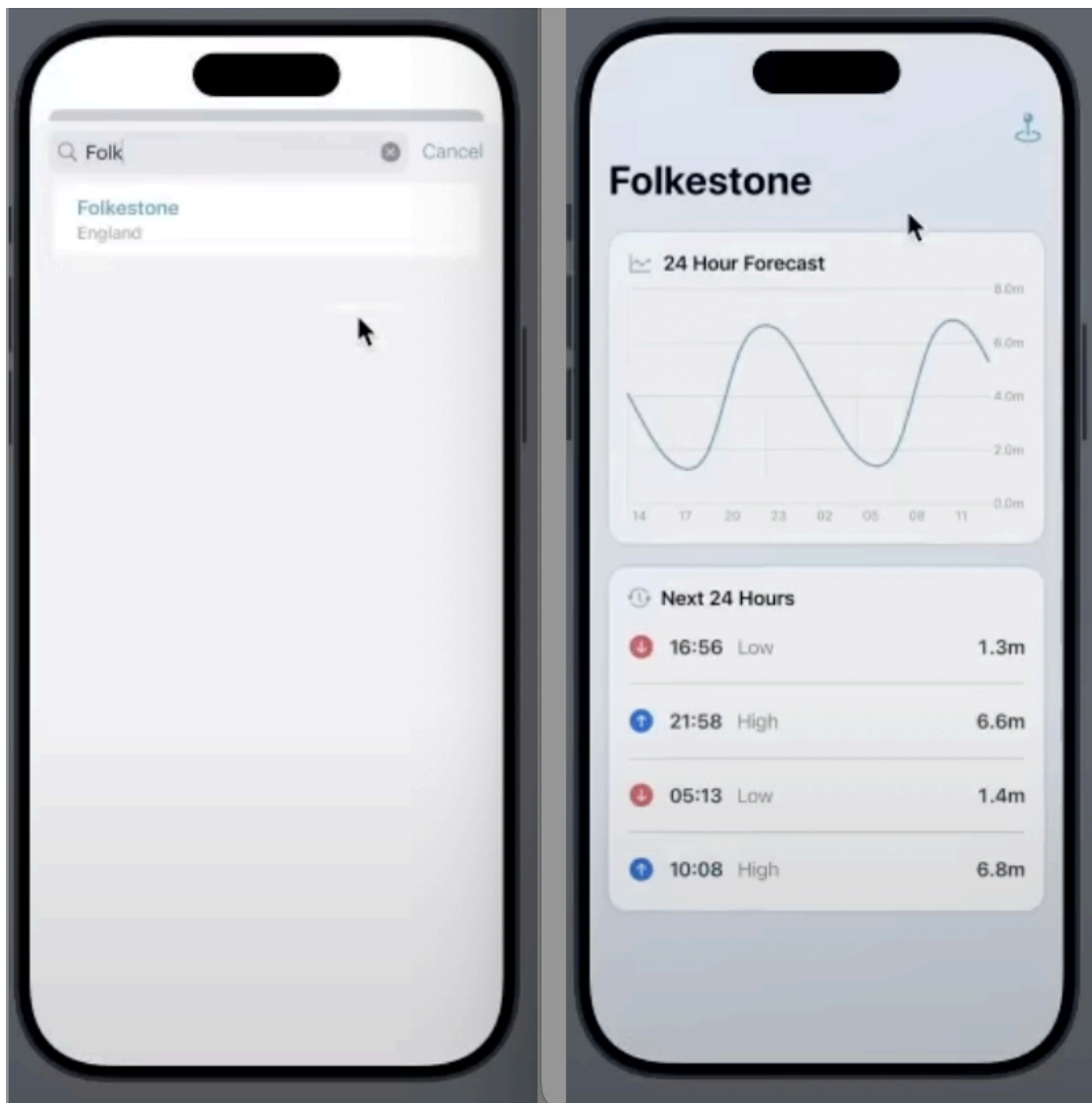


Теперь у меня есть красивое, милое приложение Tide Times в стиле Apple с графиком, показывающим приливы в течение следующих 24 часов, кроме того добавлено несколько красивых иконок.

Итак, это просто краткий обзор того, как создавать приложения для iOS.

Теперь оно выглядит совершенно фантастически.

Давайте быстро протестируем его, введя другое местоположение в поисковой строке и выберем **Folkestone** (Фолькстоун). И снова выглядит абсолютно идеально:



Теперь мы действительно довольны нашим приложением.

Мы можем протестировать приложение Tide Times на симуляторе iPhone .

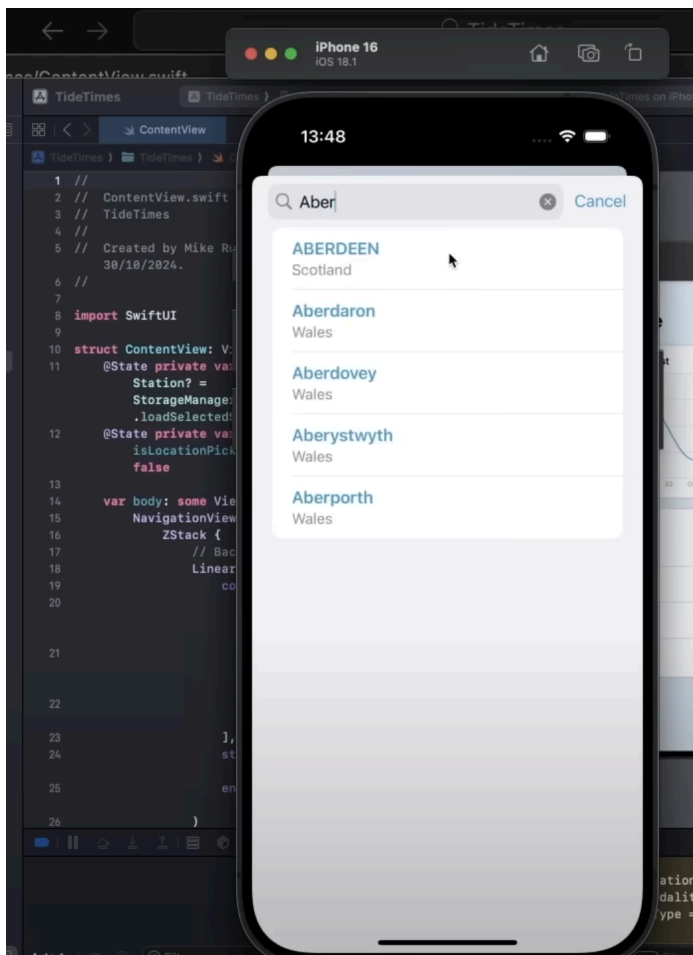
И опять я могу поискать что-нибудь. Давайте найдем Southampton (Саутгемптон).



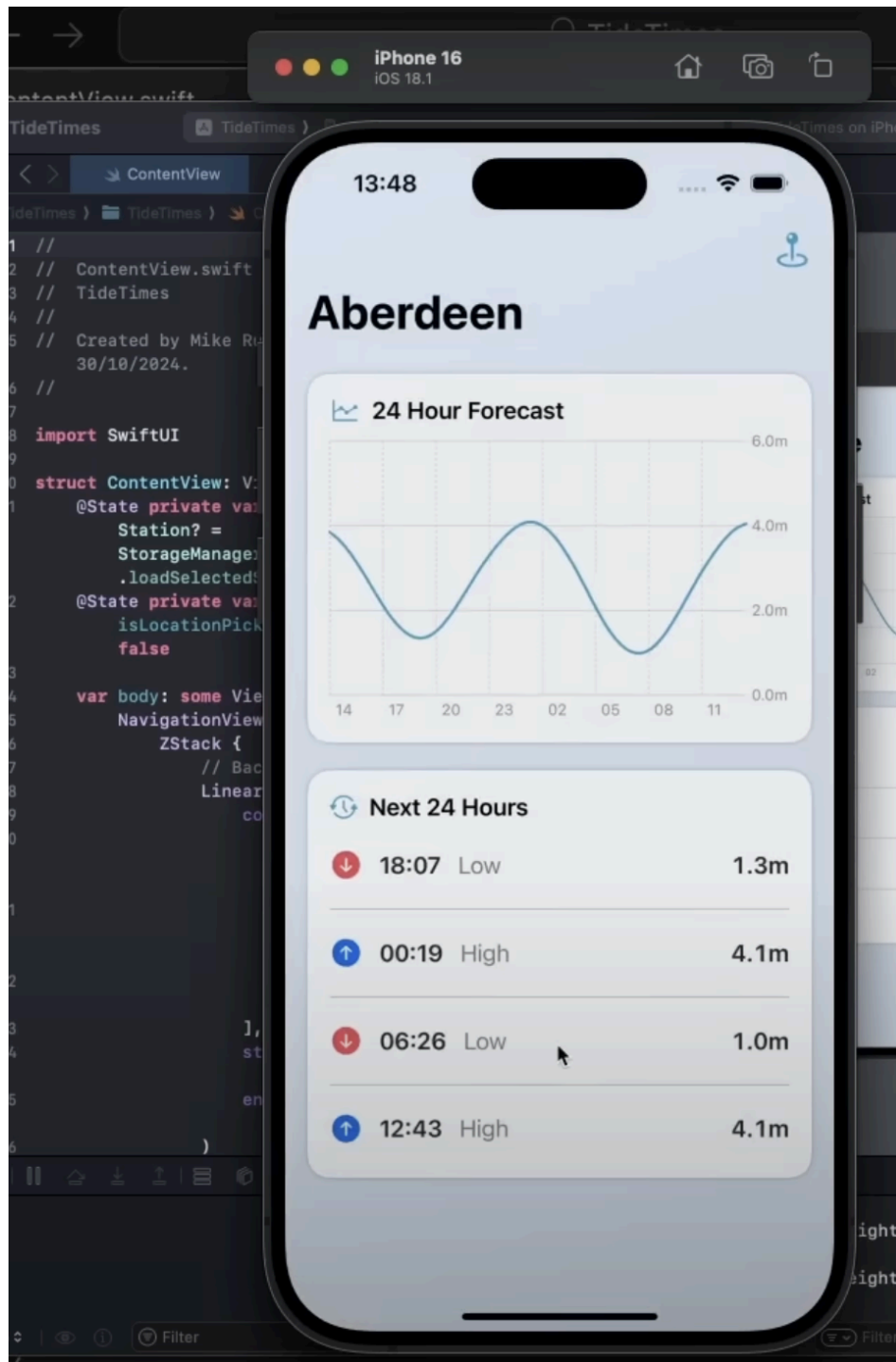
И да, у нас есть время приливов в Саутгемптоне.

Давайте найдем что-нибудь еще.

Давайте найдем Aberdeen (Абердин), например.



И да, у нас есть время приливов в Абердине:

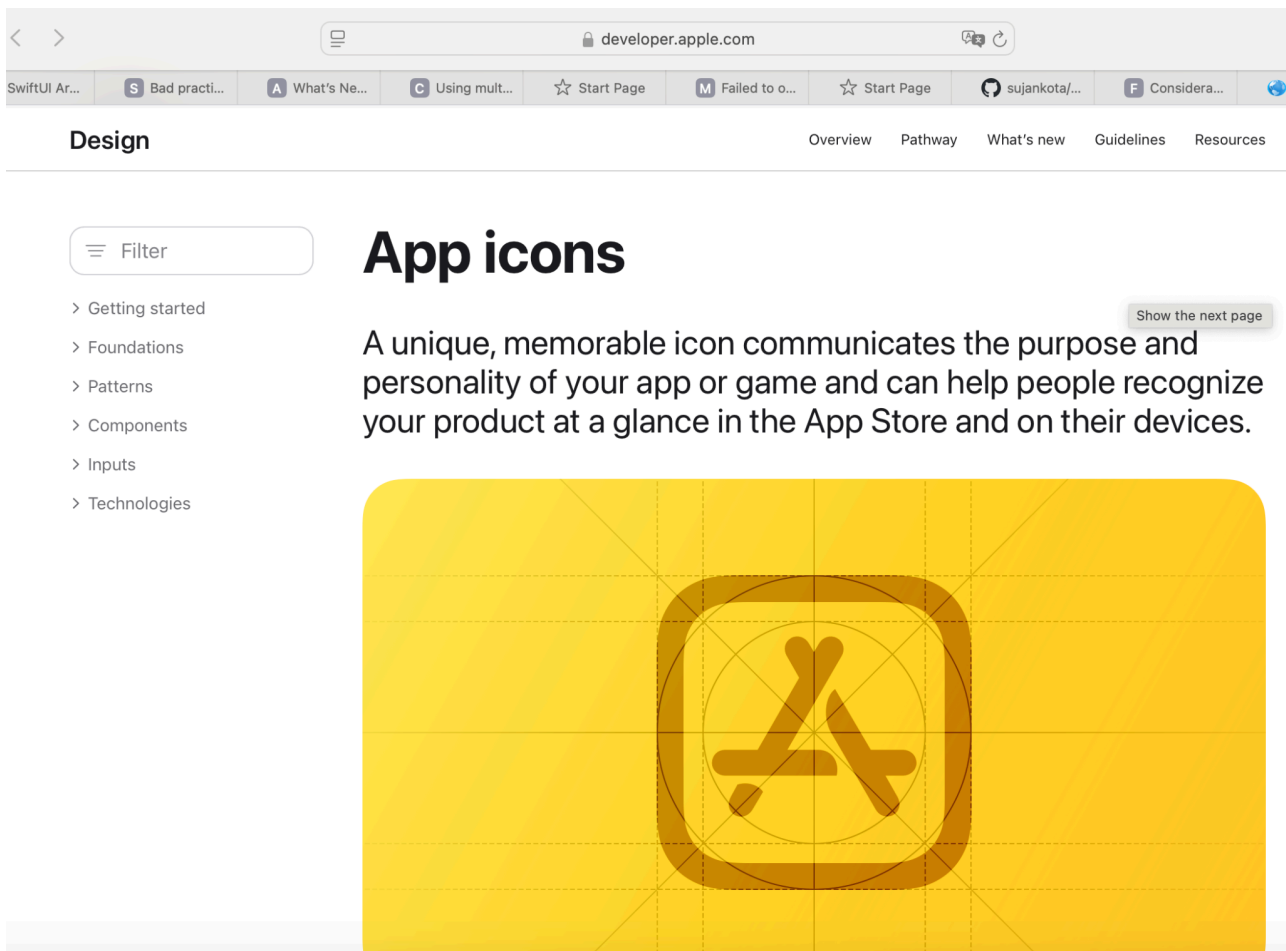


Красивое приложение, которым, возможно, гордилась бы **Apple**.

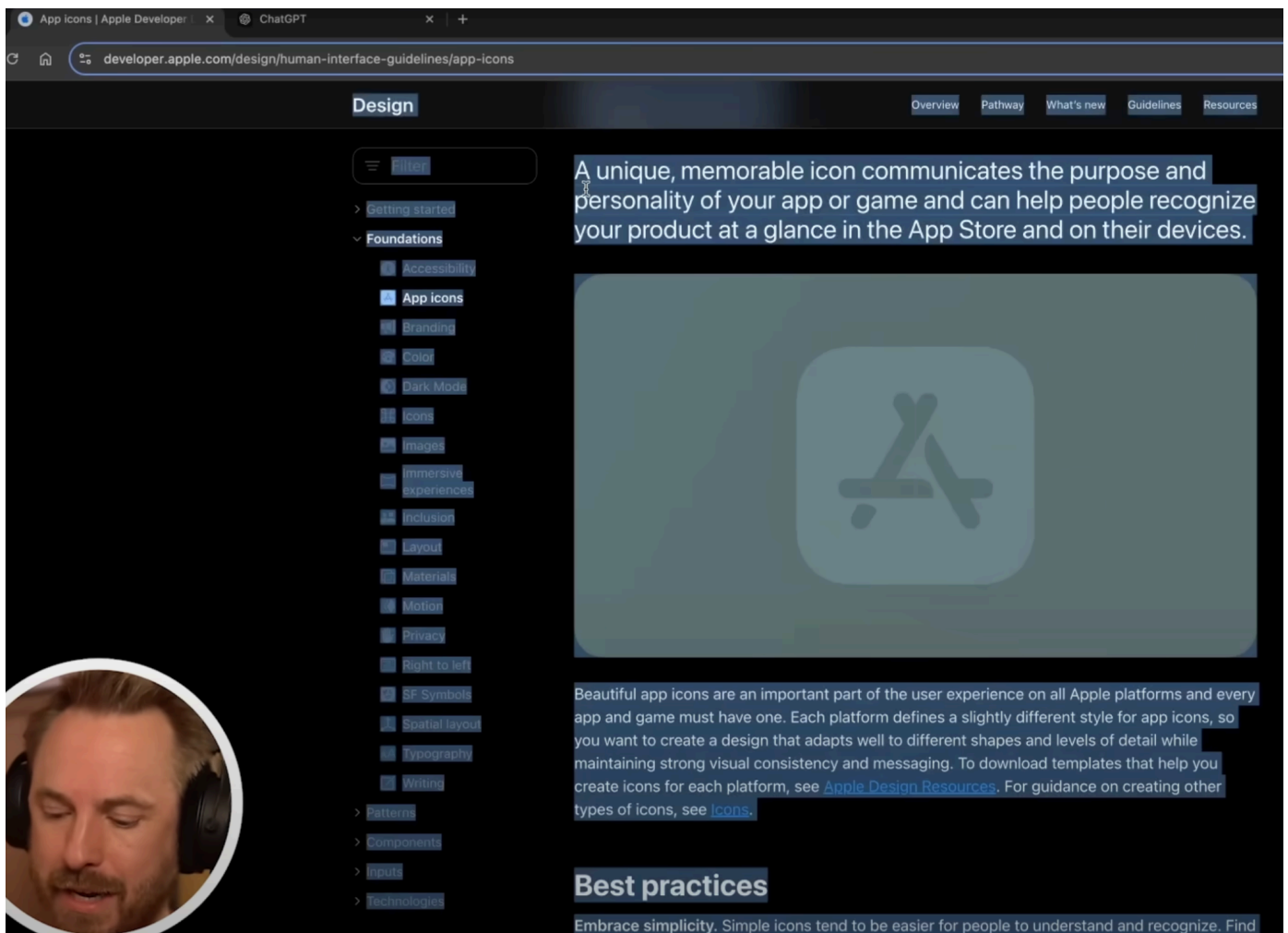
Далее, давайте создадим иконку приложения и загрузим его в приложение, Создадим иконку приложения с помощью инструментов генерации изображений ИИ, чтобы мы могли загрузить его на мое устройство и протестировать в реальных условиях.

Итак, вы, возможно, помните, что ранее я открыл эту веб-страницу от Apple, которая дает рекомендации разработчикам по созданию иконок приложений

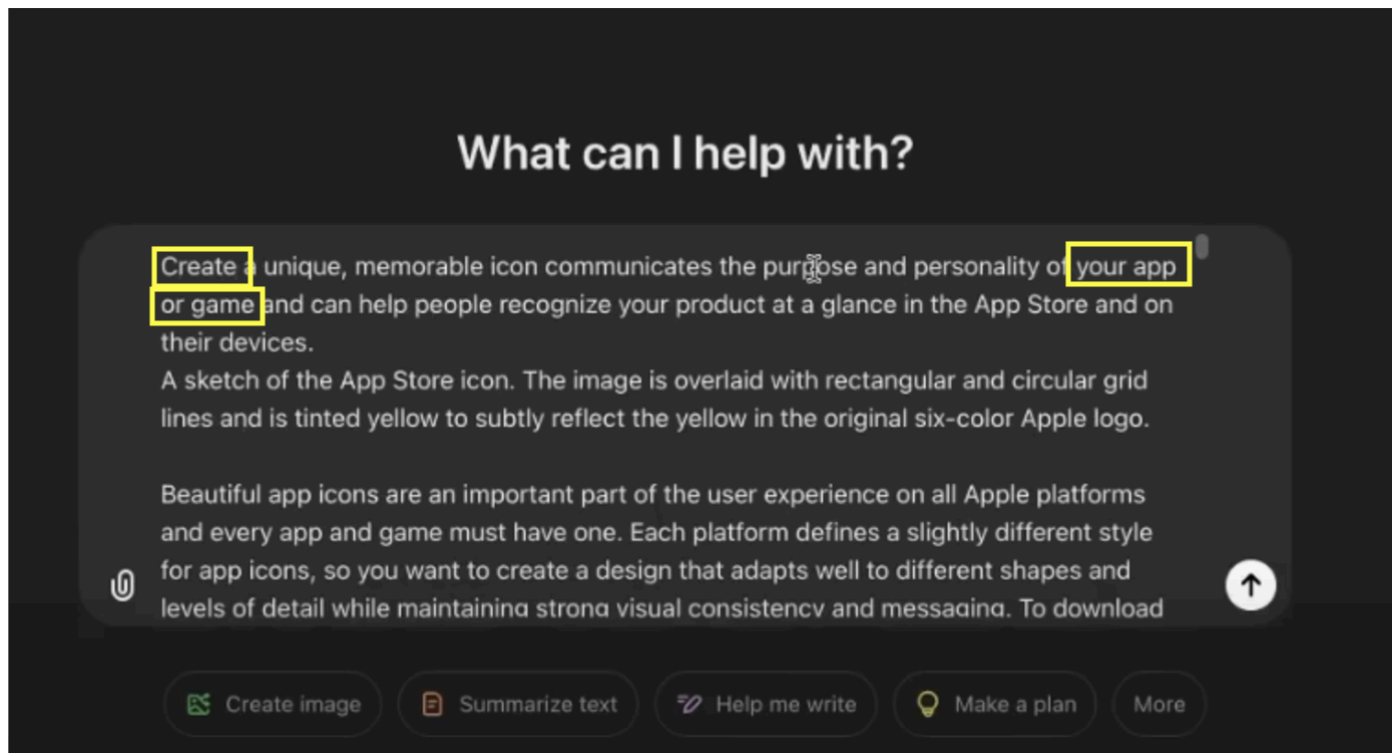
<https://developer.apple.com/design/human-interface-guidelines/app-icons> .



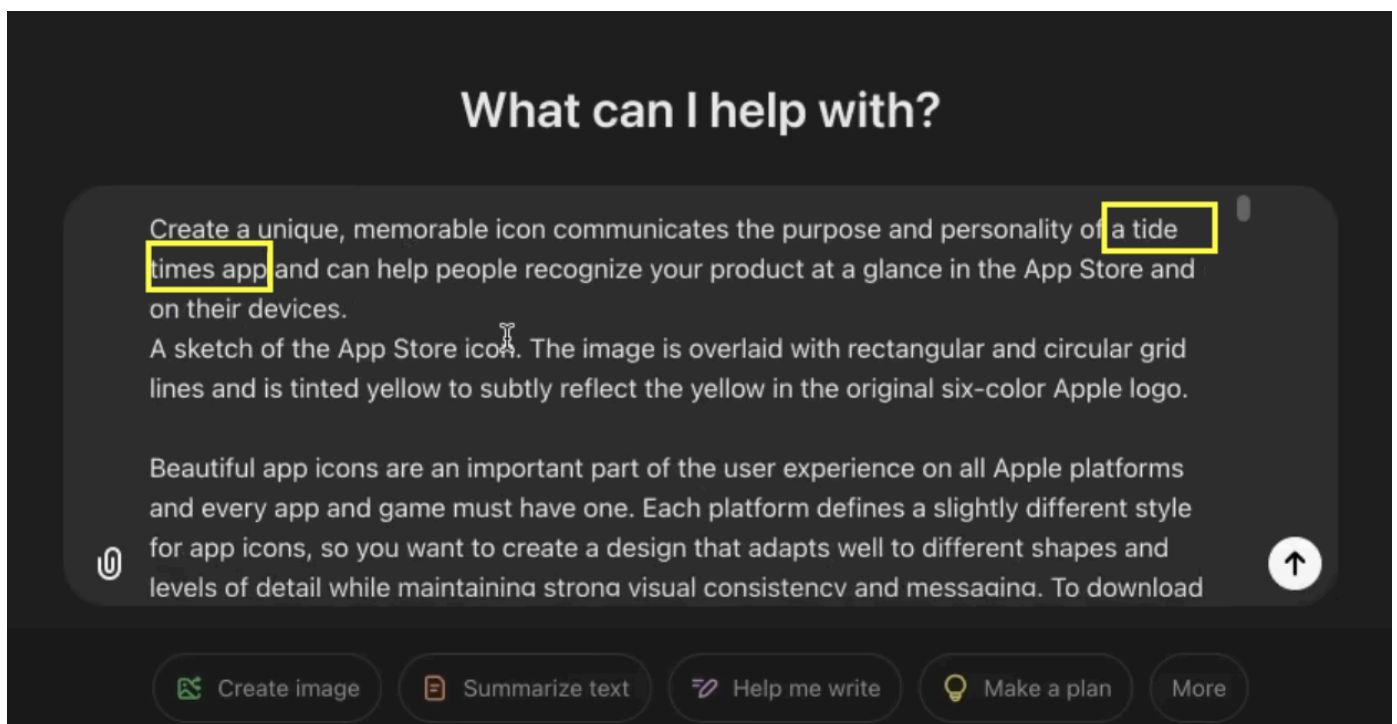
Что ж, давайте нажмем **command +A** на этой странице, чтобы скопировать всю информацию.



Теперь вставим это в ChatGPT и немного изменим его:



Добавим текст : “**Создать** уникальный, запоминающийся значок, который передает цель и индивидуальность....”, а затем мы просто вместо “your app or game” добавим сюда **tide times app** (приложение Tide Times):



весь текст внизу остается без изменения.

Мы оставим все как есть и позволим **Dall-E** делать свое дело внутри **ChatGPT**.

Сейчас вы увидите.

Он просто возвращает мне текст о том, как должна выглядеть иконка.

- **Gradient Background:** The background gradient would transition vertically, starting with a brighter blue at the top, fading into a dusky purple or golden hue at the bottom, reflecting the natural changes in tide and light over the day.
- **Yellow Accents:** Adding a soft, subtle yellow reflection on the wave or near the top edge of the icon to suggest sunlight or moonlight enhances the look. This yellow hue is subtle but ties back to the classic Apple palette, making the app feel familiar.

Design Variants:

1. **Light Icon:** A bright, clean version with blue and light orange gradients, keeping the wave design prominent.
2. **Dark Icon:** The colors deepen to a darker blue and violet background, with a more subdued wave to enhance visibility on dark mode.
3. **Tinted Icon:** A grayscale rendition for Apple's tinted mode, where the wave transitions through varying shades of gray on a subtle dark gradient background.

Icon Layout:

On the App Store, this icon would appear within Apple's typical rounded-square grid, making it adaptable for all Apple device platforms. The **circular and rectangular grid** ensures that the central wave design is easily visible and uncluttered, helping people instantly recognize it as a tide-related app.

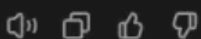
The icon's adaptability for light, dark, and tinted settings would ensure it stands out while feeling right at home on any Apple device.

Сейчас скажу создать эту иконку:

Icon Layout:

On the App Store, this icon would appear within Apple's typical rounded-square grid, making it adaptable for all Apple device platforms. The **circular and rectangular grid** ensures that the central wave design is easily visible and uncluttered, helping people instantly recognize it as a tide-related app.

The icon's adaptability for light, dark, and tinted settings would ensure it stands out while feeling right at home on any Apple device.



Create that icon

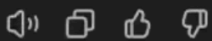
И бум... **Dall-E** здесь работает.

Он создал иконку моего приложения.



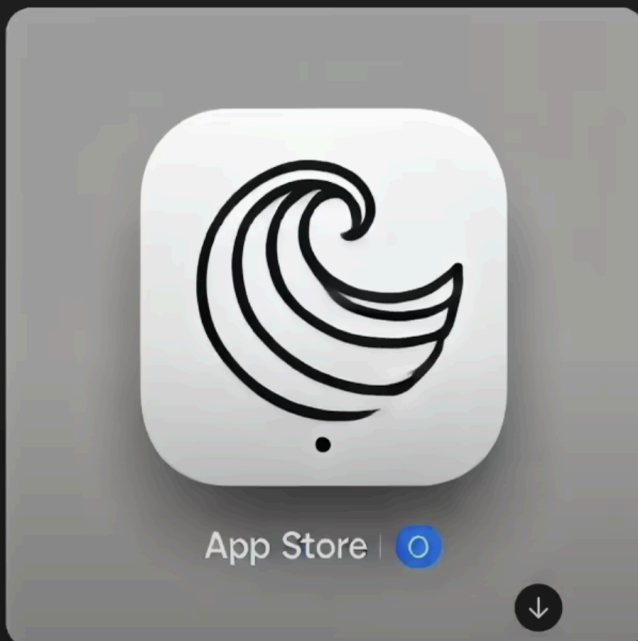
Here's the icon for your tide times app, featuring the stylized wave and gradient to convey a dynamic yet clean aesthetic, aligning with the principles for memorable and adaptable app design. Let me know if you'd like any adjustments.

Я снова попрошу создать иконку без границ:



Again, no borders or background. full screen

Теперь, несмотря на многократные подсказки:



Again, no borders or background. full screen



↗

Here's the updated icon, filling the entire frame with the wave design and gradient background. This version emphasizes the tide theme with a seamless, full-screen approach. Let me know if you'd like any other tweaks!



↗

Here's the iOS app icon designed with the minimalist wave graphic and day-to-night gradient, as described. It's optimized for the iOS aesthetic with rounded edges and a clean, modern look. Let me know if further refinements are needed!



Я не мог заставить его вырезать границу, кроме одного раза здесь.



Here's the updated icon, filling the entire frame with the wave design and gradient background. This version emphasizes the tide theme with a seamless, full-screen approach. Let me know if you'd like any other tweaks!



Каждый раз, когда он давал мне прототип с экраном iPhone или рамкой, **Dall-E** был не лучшим:

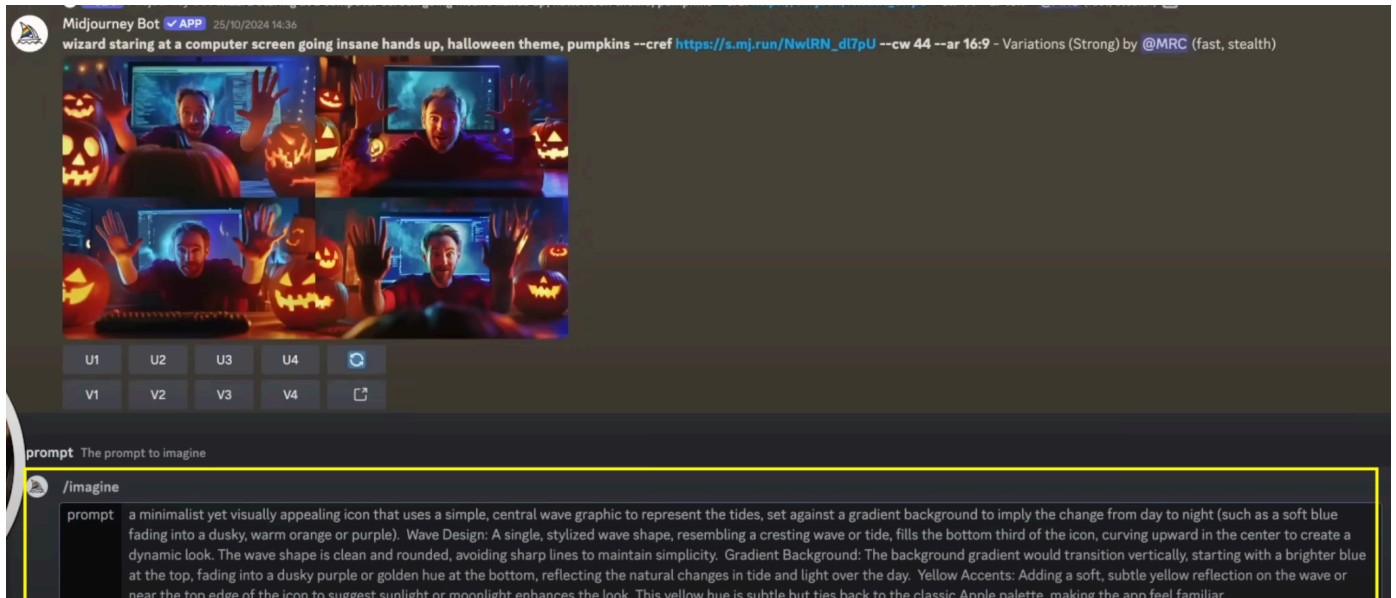


Here's the iOS app icon designed with the minimalist wave graphic and day-to-night gradient, as described. It's optimized for the iOS aesthetic with rounded edges and a clean, modern look. Let me know if further refinements are needed!



Поэтому я скопировал подсказку из , **Dall-E** и перешел в **MidJourney**.

Здесь, в **MidJourney** я использую команду **/imagine** и вставляю ее в подсказку **prompt**, чтобы создать дизайн иконки моего приложения:



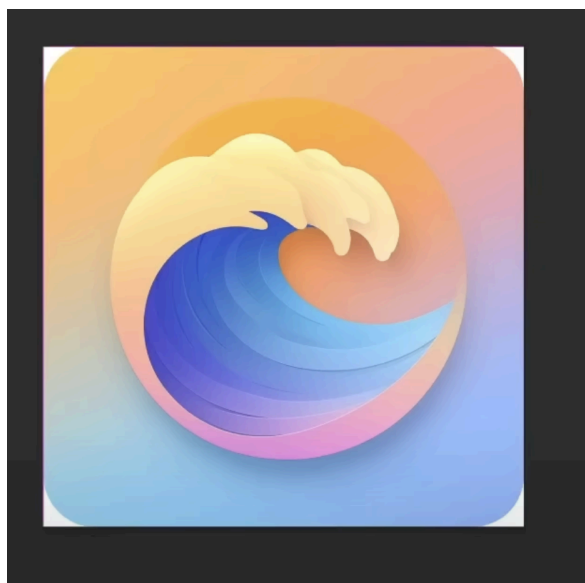
И бум. Через несколько мгновений у нас было несколько пригодных для использования дизайнов:



Мне очень нравится их дизайн раньше, поэтому я собираюсь увеличить масштаб.



Генераторам изображений, похоже, нравится добавлять изогнутые края вокруг, я собираюсь выбрать это изогнутое изображение



и использовать генеративную заливку (**generative fill**) в **Photoshop**.
И бум!

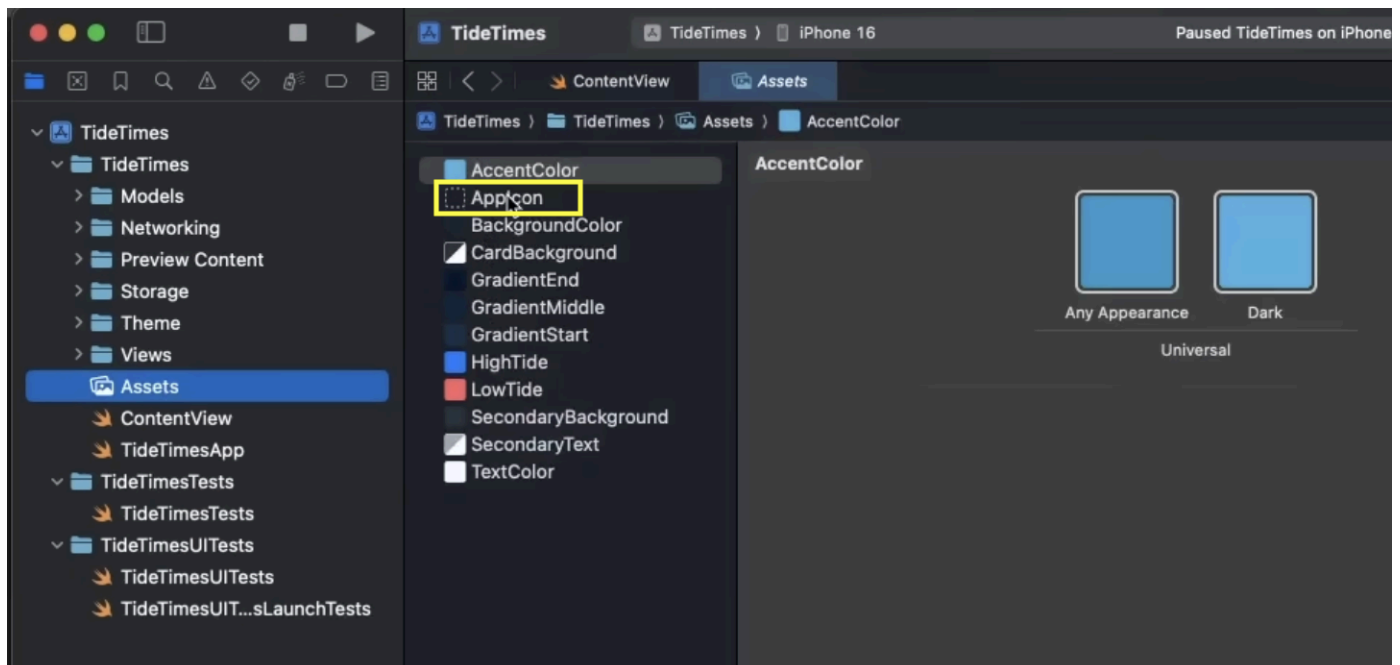


Теперь у нас есть красивая иконка приложения, которую мы можем использовать.

Добавление иконки в проект Xcode

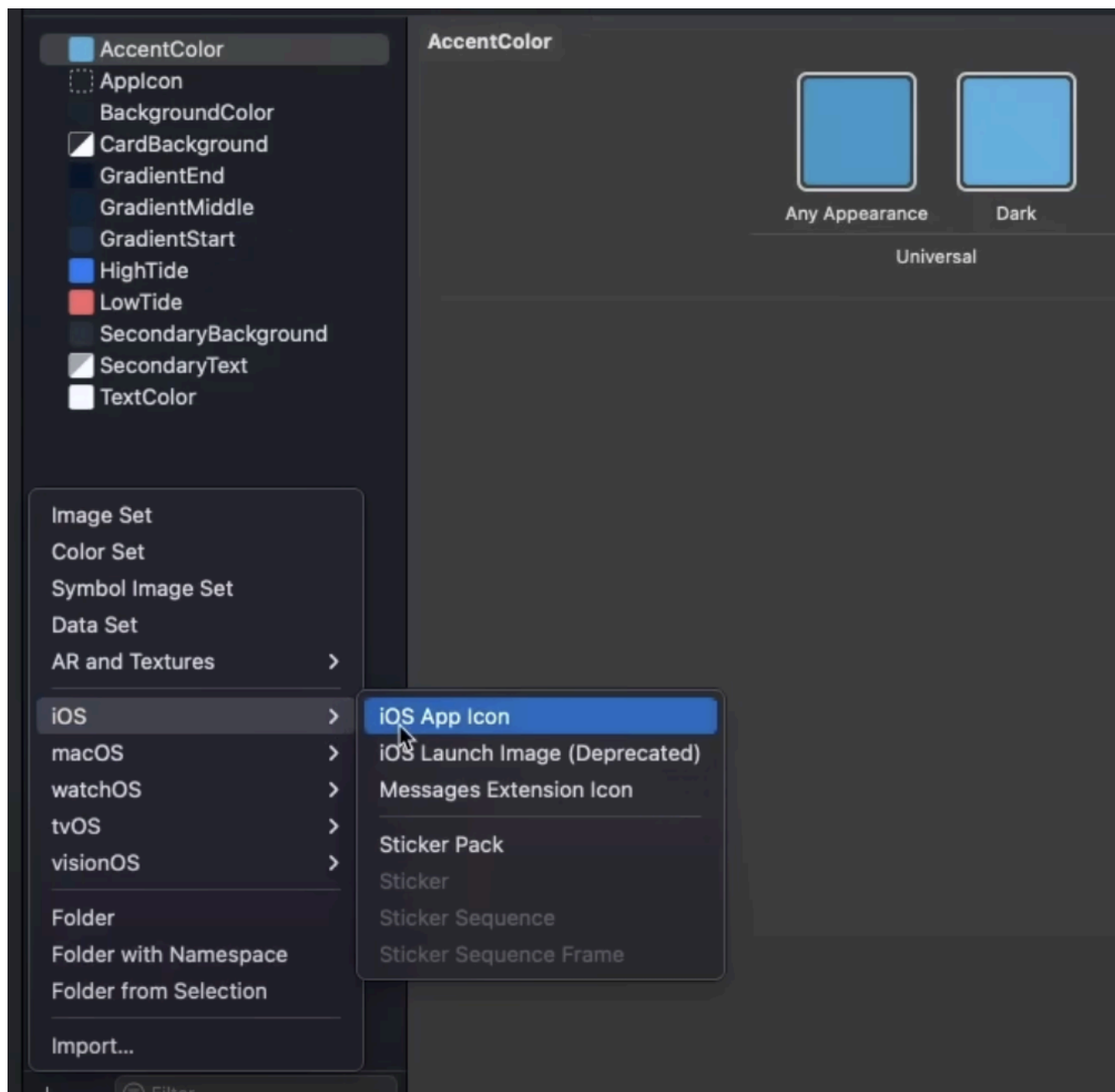
Теперь вернемся в **Xcode** и перейдем к ресурсам **assets** моего приложения.

И вы увидите, что здесь уже есть ресурс значка приложения **App Icon**.

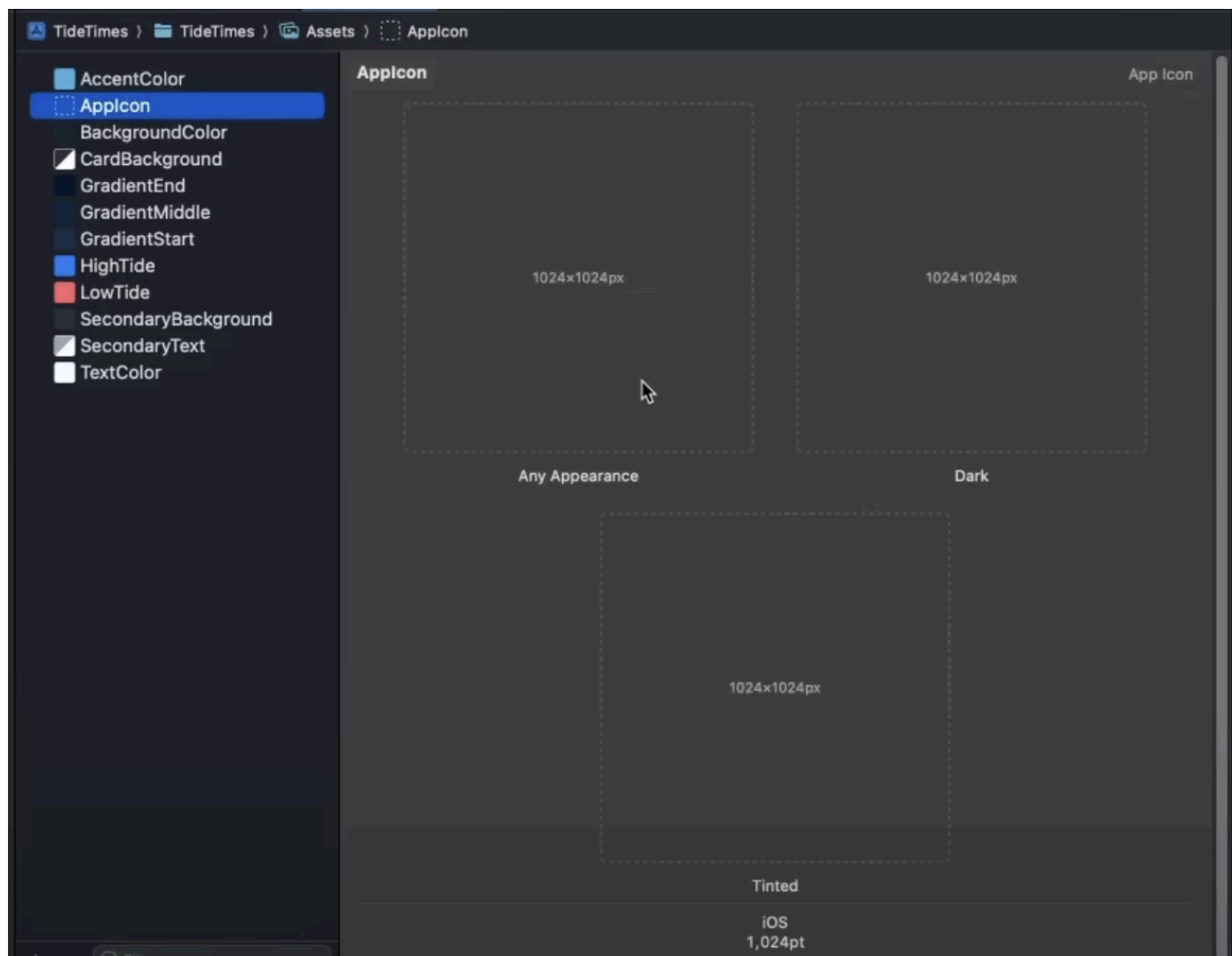


Если его нет, не волнуйтесь.

Просто нажмите кнопку «плюс» здесь. Перейдите в iOS и найдите значок приложения iOS.

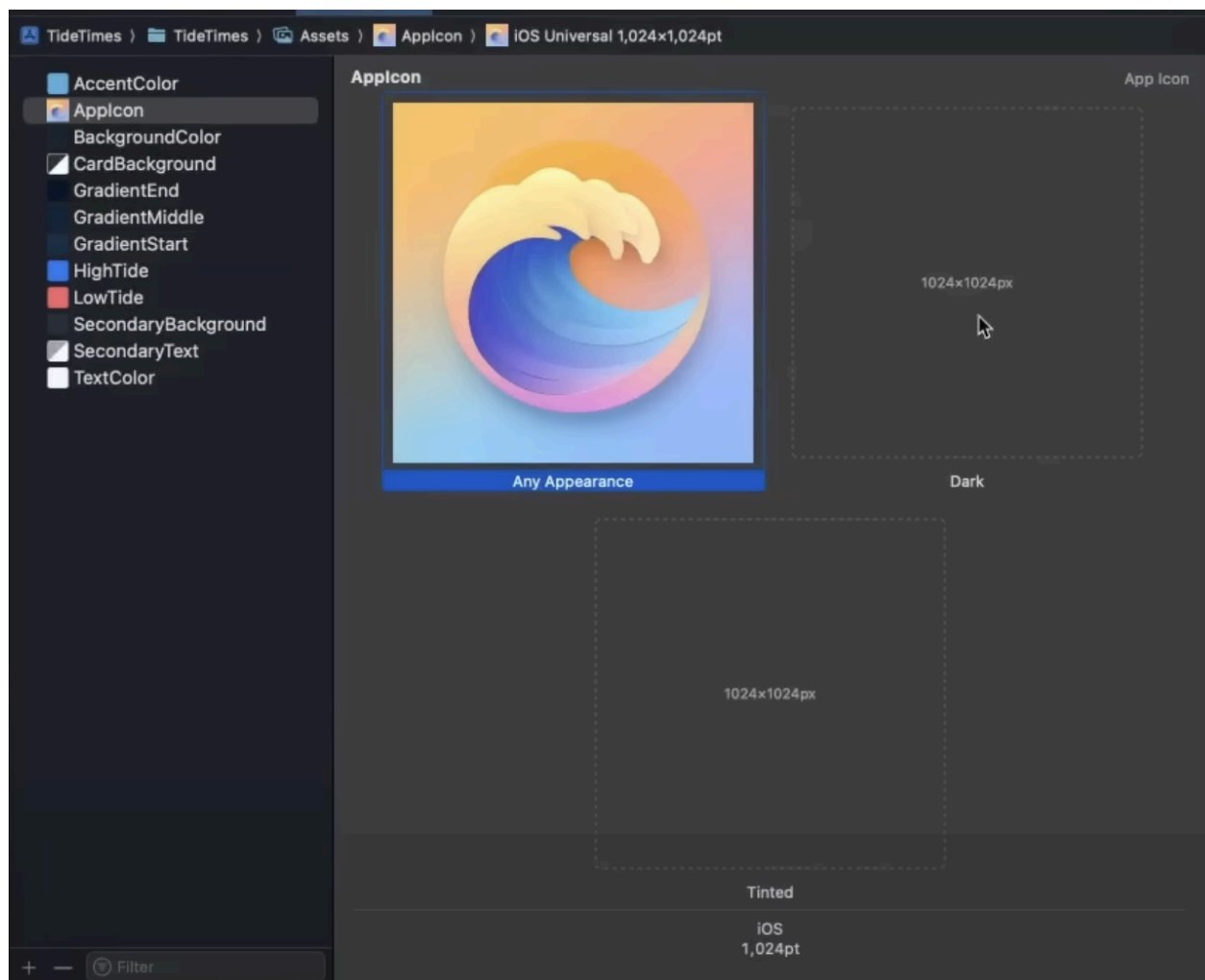


Но поскольку он у меня уже есть, мне просто нужно нажать здесь и загрузить его теперь. У меня есть здесь красивые разные варианты.

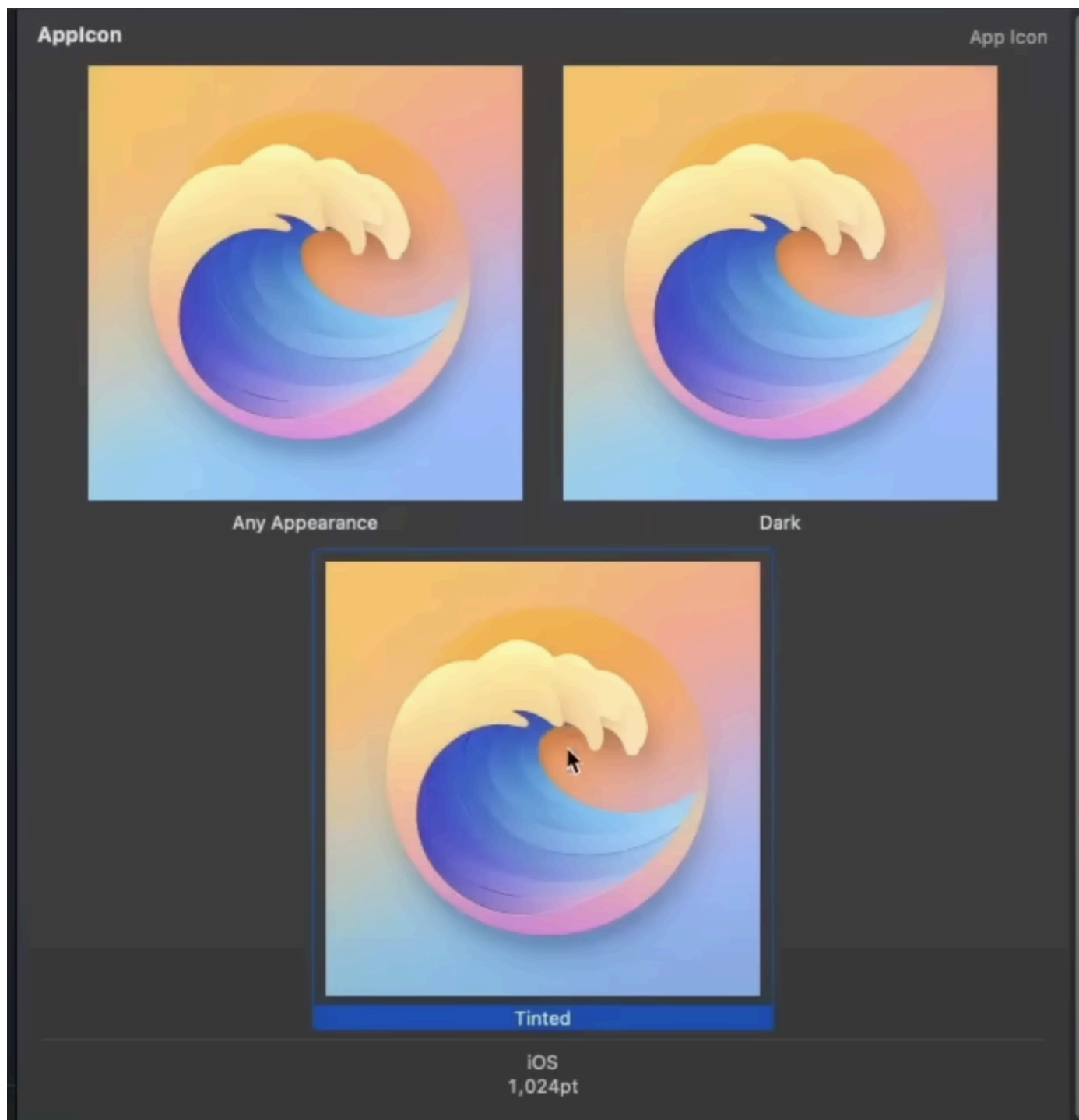


У меня есть любой внешний вид **Any Appearance** темный **Dark** и тонированный **Tinted**.

Мы просто начнем **Any Appearance**, и я загрузу свою иконку, которая находится на моем рабочем столе.



Сейчас я просто поставлю эту иконку в каждый дизайн, который смогу.



Конечно, если бы это было производственное приложение, изменил бы их так, чтобы они были абсолютно правильными.

11:09

Но пока это будет работать.

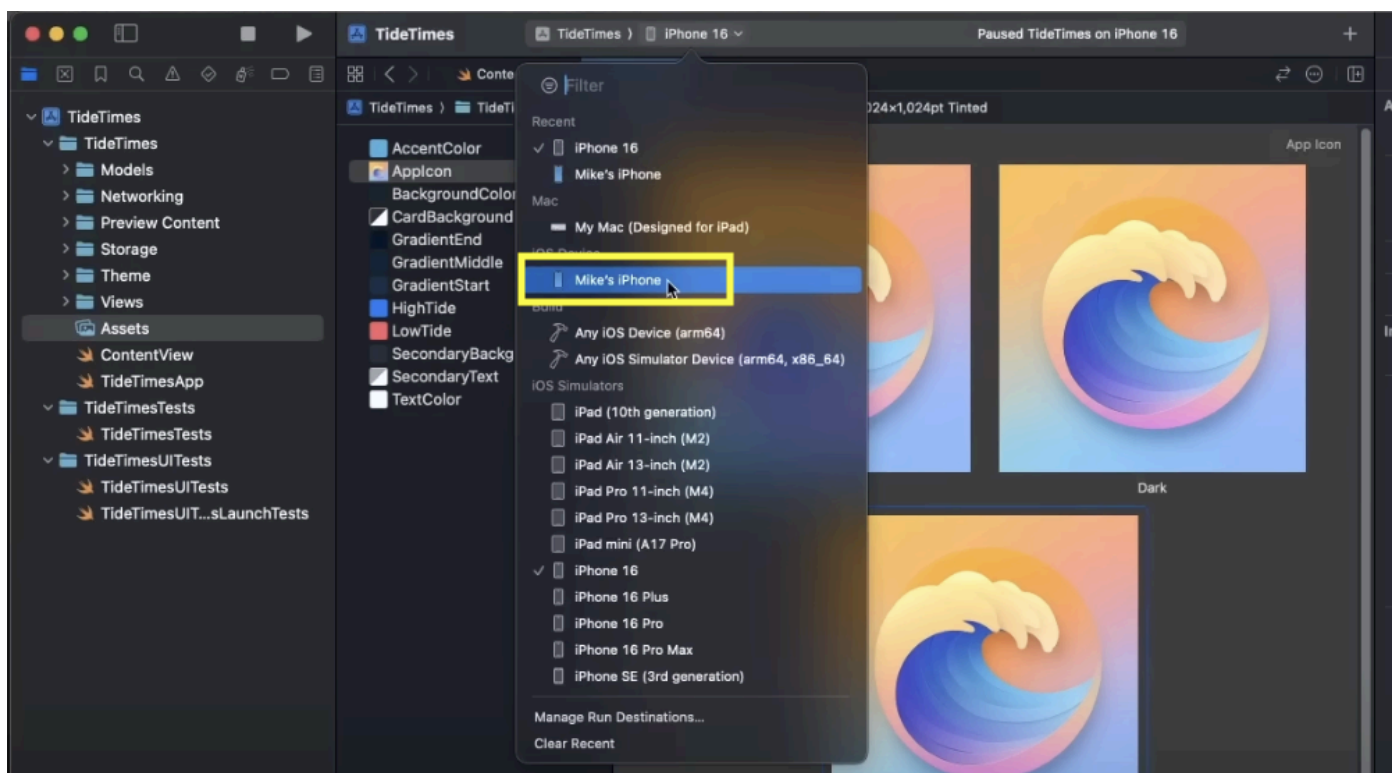
Установка приложения на настоящем iPhone

Сейчас самое время загрузить его на мой **iPhone** и опробовать в реальном мире.

Чтобы сделать это, нам понадобится кабель для подключения телефона к моему **Mac**.



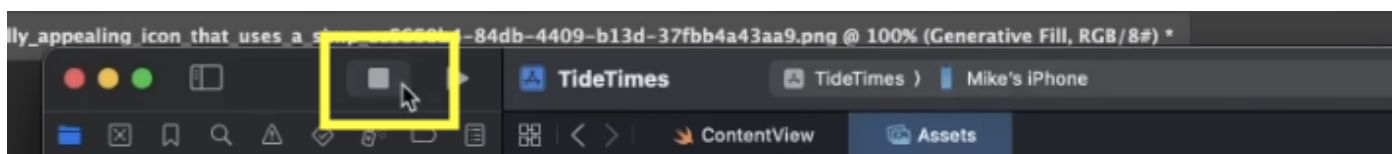
Когда я сделаю это, в **Xcode** я выберу в выпадающем списке свой iPhone, подключенный к моему компьютеру.



Кстати, вам нужно ответить на своём телефоне, что вы доверяете компьютеру.

Иначе он не отобразится. iPhone Майка.

И я нажму кнопку остановки, чтобы остановить виртуальный симулятор, который мы запустили ранее



и нажмем на кнопку воспроизведения, чтобы загрузить это на мой iPhone.



Сборка прошла успешно.

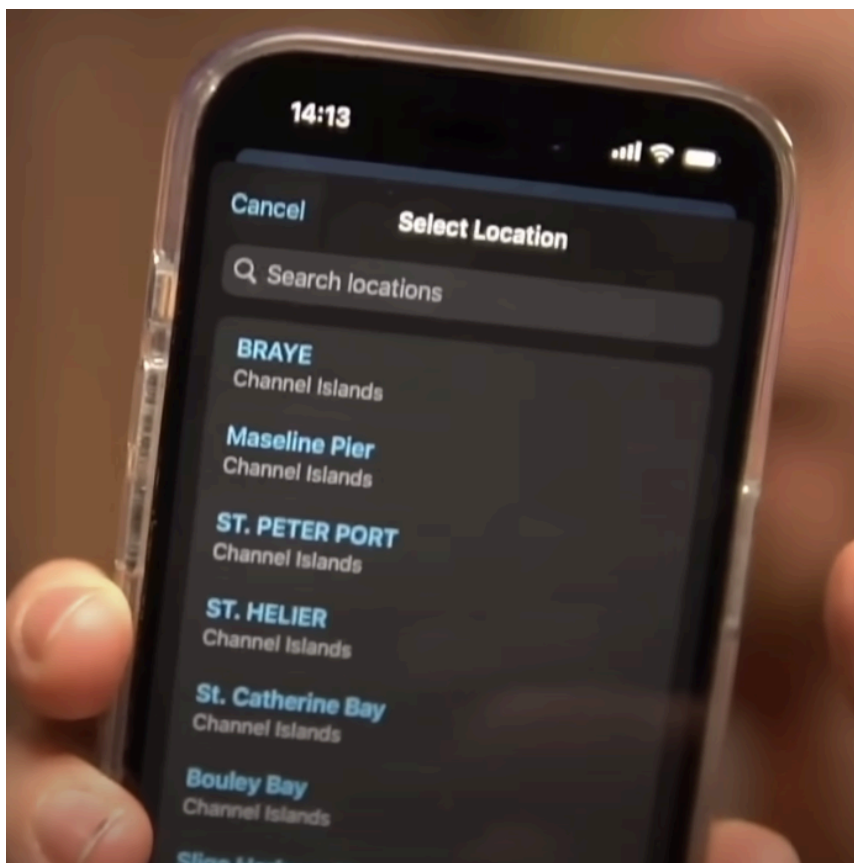
И теперь он отправляется на мой телефон и устанавливает приложение.

После этого вы должны иметь возможность открыть приложение.



Выглядит очень хорошо, не так ли?

Затем я могу нажать на отметку местоположения в верхнем правом углу и могу выбрать местоположение.



Давайте просто выберем любое местоположение и вуаля!



Теперь приложение полностью функционально на моем iPhone.

Это невероятно.

И снова я тестирую разные местоположения. Они все работают хорошо.

А еще, поскольку мой телефон находится в темном режиме, он даже следует схеме темного режима внутри приложения, что круто.

Всего несколько минут вашего времени, один ИИ инструмент, и у вас есть полностью функционирующее приложение iOS на вашем физическом устройстве.

Теперь имейте в виду, если вы используете бесплатную учетную запись разработчика Apple, вы можете использовать приложение в течение семи дней, но затем вам нужно будет перезагрузить его через Xcode, чтобы оно снова заработало.

В противном случае срок его действия просто истечет.

Если вы заплатите за учетную запись разработчика Apple, которая стоит 99 долларов в год, вы сможете использовать такие приложения на своем телефоне в течение одного года, прежде чем вам понадобится их продлевать, и это также дает вам возможность опубликовать свое приложение в App Store, что, я полагаю, и является конечной целью.

Сейчас самое время донести идею, которая у вас в голове, до всего мира и до каждого на устройстве iOS.

Я буду поддерживать вас в комментариях, так что дайте мне знать о ваших вопросах и мыслях.

Мне не терпится увидеть, что вы создадите, и присоединиться к моему сообществу, где мы обсуждаем разработку приложений для iOS на основе ИИ.

Кроме того, сейчас существует большое сообщество, в котором обсуждаются различные аспекты этого и других инструментов без кода. Так что присоединяйтесь к нам.

.

.

.

.