

# Initialize Frontend

Now it's time to work on our Frontend. We will be using Next JS which is a popular react Framework used by many Web3 projects for frontends.

In your command prompt tool of choice navigate to any directory of choice and create your Next JS application using the [Create Next App tool](#).

With NPM:

`npx create-next-app@latest` With Yarn:

`yarn create next-app` This is how we answered the Propts when creating the application:

What is your project named? based-aa Would you like to use TypeScript? Yes Would you like to use ESLint? Yes Would you like to use Tailwind CSS? No Would you like to use `src/` directory? Yes Would you like to use App Router? (recommended) No Would you like to customize the default import alias? Yes info Although the App router is mentioned as the preferred way of using Next, the pages router is much more stable at the time of writing this tutorial. We recommend saying no when promoted to use the App router. You can follow the prompts to receive the scaffolding for your React application. You can use any of the package managers of your choice, but to keep things simple we will be using Yarn from this point.

Install the following dependencies:

`yarn add @biconomy/account @biconomy/modules ethers@5.7.2 react-toastify` Additionaly to help with some polyfill errors we will need to update `ournext.config.js` file which is located at the root of our project. Copy the code below and replace the current contents of the file:

```
/** @type
{ import ( 'next' ) . NextConfig } */ const nextConfig =
{ reactStrictMode :
true , webpack :
( config ,
{ isServer } )
=>
{ if
( ! isServer )
{ config . resolve . fallback
=
{ fs :
false , net :
false , tls :
false , } ; } return config ; } , } ;
module . exports
```

`= nextConfig` ; Lastly we're going to replace the default styles with some prewritten styles mentioned below. Similar to the smart contract section, we won't focus much on the CSS side of things but this will give you some basic layouts and center all of your content in the middle of the page.

Replace `theglobal.css` and `theHome.module.css` with the styles below.

```
global.css :root
{ font-family : Inter , system-ui , Avenir , Helvetica , Arial , sans-serif ; line-height :
1.5 ; font-weight :
400 ;
```

color-scheme : light dark ; color :

rgba ( 255 ,

255 ,

255 ,

0.87 ) ; background-color :

**242424**

;

font-synthesis : none ; text-rendering : optimizeLegibility ; -webkit-font-smoothing : antialiased ; -moz-osx-font-smoothing : grayscale ; -webkit-text-size-adjust :

100 % ; }

a

{ font-weight :

500 ; color :

**646cff**

; text-decoration : inherit ; } a :hover

{ color :

**535bf2**

; }

body

{ margin :

0 ; display : flex ; place-items : center ; min-width :

320 px ; min-height :

100 vh ; }

h1

{ font-size :

3.2 em ; line-height :

1.1 ; }

button

{ border-radius :

8 px ; border :

1 px solid transparent ; padding :

0.6 em

1.2 em ; font-size :

1 em ; font-weight :

500 ; font-family : inherit ; background-color :

# 1a1a1a

```
; cursor : pointer ; transition : border-color 0.25 s ; } button :hover  
{ border-color :
```

# 646cff

```
; } button :focus , button :focus-visible  
{ outline :  
4 px auto -webkit-focus-ring-color ; }  
@media  
( prefers-color-scheme : light )  
{ :root  
{ color :
```

# 213547

```
; background-color :
```

# ffffff

```
; } a :hover  
{ color :
```

# 747bff

```
; } button  
{ background-color :
```

# f9f9f9

```
; } } Home.module.css
```

# root

```
{ max-width :  
1280 px ; margin :  
0 auto ; padding :  
2 rem ; text-align : center ; }  
.logo  
{ height :  
6 em ; padding :  
1.5 em ; will-change : filter ; transition : filter 300 ms ; } .logo :hover  
{ filter :  
drop-shadow ( 0
```

0

2 em

## 646cffaa

```
) ; } .logo .react :hover
```

```
{ filter :
```

```
drop-shadow ( 0
```

0

2 em

## 61dafbaa

```
) ; }
```

```
@keyframes logo-spin
```

```
{ from
```

```
{ transform :
```

```
rotate ( 0 deg ) ; } to
```

```
{ transform :
```

```
rotate ( 360 deg ) ; } }
```

```
@media
```

```
( prefers-reduced-motion : no-preference )
```

```
{ a :nth-of-type ( 2 )
```

```
.logo
```

```
{ animation : logo-spin infinite 20 s linear ; } }
```

```
.card
```

```
{ padding :
```

```
2 em ; }
```

```
.read-the-docs
```

```
{ color :
```

## 884c30

```
; }
```

```
.linkWrapper
```

```
{ margin-top :
```

```
15 px ; }
```

```
.read-the-docs :hover
```

```
{ color :
```

## b84814

; transition :

0.7 s ; }

.demoButton

{ background-color :

**884c30**

; color :

**fff**

; margin-bottom :

5 px ; }

.demoButton :hover

{ background-color :

**b84814**

; transition :

0.7 s ; }

.viewNFT

{ color :

**b84814**

; }

**root**

{ max-width :

1280 px ; margin :

0 auto ; padding :

2 rem ; text-align : center ; }

.logo

{ height :

6 em ; padding :

1.5 em ; will-change : filter ; transition : filter 300 ms ; } .logo :hover

{ filter :

drop-shadow ( 0

0

2 em

**646cffaa**

```
) ; } .logo .react :hover  
{ filter :  
drop-shadow ( 0  
0  
2 em
```

## 61dafbaa

```
) ; }  
@keyframes logo-spin  
{ from  
{ transform :  
rotate ( 0 deg ) ; } to  
{ transform :  
rotate ( 360 deg ) ; } }  
@media  
( prefers-reduced-motion : no-preference )  
{ a :nth-of-type ( 2 )  
.logo  
{ animation : logo-spin infinite 20 s linear ; } }  
.card  
{ padding :  
2 em ; }  
.readDocs  
{ color :
```

## ff4e17

```
; }  
.linkWrapper  
{ margin-top :  
15 px ; }  
.readDocs :hover  
{ color :
```

## b84814

```
; transition :  
0.7 s ; }  
.demoButton  
{ background-color :
```

**ff4e17**

; color :

**fff**

; margin-bottom :

5 px ; }

.demoButton :hover

{ background-color :

**b84814**

; transition :

0.7 s ; }

.viewNFT

{ color :

**ff4e17**

; }

.demoContainter

{ text-align : center ; width :

100 vw ; }

.main

{ text-align : center ; width :

100 vw ; }

.connect

{ background-color :

**ff4e17**

; color :

**fff**

; } Now with our frontend initialized lets start building out our integration in the next section[Previous Session Validation](#)  
[Module Next SDK Integration](#)