

ordKit_Support



- Optimizing flow for Scale
- Optimize API
- Update PKG
- API endpoints got overwhelmed

/Collections source API Collection \$8 money
 /Inscriptions

Fix St. Current proj (it's currently showing in upcoming) ✓

L > F, P, L but now in edit start/end date show EST w/ GMT+7 label

The workflow runs in evening render, not sure what we should replace it or use dependencies.

When query gets updated/changed we update at column with newest change.
 ↑ this is supabase, need access

/mint/history | Support dashboards

docs.ordinal.com

Create ticket

form

User info

minimize spam

(Verify mint b4 submit)



mintenV table
 ↓ there's a column
 (minter discord id)

Requirements

1. Discord Auth

2. ref id for mint

for connecting DB don't worry for 3. payment hash (ETH/BTC)
 now, just do auth w/ new supabase

DB, try adding (Role Based Access Control)

Then only connect DB after Smart form Complete (Connect for auth w/ ref ID)

We could check new db If user already submitted ticket?
have a local column? Error prompt if they have existing ticket?

DB Logic

User Table

User roles (customer/admin) ← Store user authentication information from discord, username, email, Discord ID, etc.
 * Involves user role, determining access level (customer or admin)
 Discord ID, email/
 ref ID?
 hash?

Ticket Table

Table for managing tickets

↳ Ticket form details

↳ Ticket (assigned admin, category, priority)

Upload ticket IMG?

Admin ticket message

respond service

Customer Table:

Ticket history,
progress,
preferences

Fks - User Table (User Auth)

ADMIN TABLE

Access to ticket Support Dashboard
 Close customer tickets
 Message customer
 Tickets Message Board

Priority Hierarchy

1. Allow users to Auth Login, Show username, Show avatar, DropdownMenu (allow Logout/Dashboard)
2. Role based access, Admin Dashboard (Show ALL TICKETS) ALLOW TICKET (Close, Reopen (If mistake), Sort by Priority, filter by Category)
3. Messaging System Admin → Customer | User History (Ticket/Message/Admin Notes (Private Notes About User))
4. Customer Dashboard → Ticket Status Check, Close ticket (CAN'T REOPEN), Respond to Admin msg., ticket history.

Buttons

Routing diff

Next12

```
/pages
└ index.tsx
  dashboard.tsx
```

Next13

```
/app
  pages.tsx (file)
  dashboard (folder)
```

OR
pages.tsx (file)

Dynamic Routing

```
/posts/
  [postId] → passed to page.tsx
    Page.tsx ←
```

Server Client
Next13 vdom → vdom

```
/app
  ↳ (auth)
  ↳ login
  ↳ pages.tsx => /Login route
```

* This is a server component, need to opt into Client Side OR runs on server

* Specify
"use client"
then render Client side

use Server Side: Fetch data, access backend directly, sensitive data,
large dependencies,
use Client Side: Interactivity (onClick, onChange), States, browser API's, custom Hooks, Class Components

* Try to do as much as possible on server side as it doesn't end up in Client-Side bundle.
Ship less to client.

Next.js Layout

```
/app
  Page.tsx
  layout.tsx → root layout
  dashboard
    Page.tsx
```

Ordkit_Support Layout



app

head.tsx → Metadata

Layout.tsx → RootLayout (Can't use SSR here, so use nested layout)

→ page.tsx → Import nested layout here



Components

Layout.tsx → import footer.tsx/navbar.tsx components here [Then import in root page.tsx]

page.tsx



footer

footer.tsx



header
navbar.tsx

* ChakraUI not working b/c doesn't work in SS components.

In RootLayout (Perform 1st to enable ChakraUI)
1. Use Client
2. Wrap {children} w/ ChakraProvider

* Bug: Dark Theme not working

↳ Bug Fix: Theme saves in DevTools → Application → Storage → Clear site data

↑ TODO: ADD to FAQ PAGE

AUTH

Supabase Auth w/ Nextjs (APP ROUTER)

Next's Auth helper pkg store session IN cookie
instead of local storage (making it available across client/Server)

Middleware runs immediately before each route rendered *

Middleware is used to refresh user session before loading server component routes

⑨ ↪ Direct type imports

AUTH -

Supabase Auth w/ Next's (APP ROUTER)

Next's Auth helper plig Store session IN Coo
instead of local storage (making it available
across (client/Server))

TO DO: Fix Discord Auth (dropouts/
Add Sign-out)
1. Create User / ticket-support (table)

2. Display ticket info for user on
their dashboard

3. Role based access

↳ Admin panel for displaying
all support tickets