

## Day 1

### DAY :- 1

### Laying the Foundation For your Marketplace Journey

#### Marketplace Type :-

#### Business Goals

Q. What problem does your marketplace aim to solve?

my marketplace aim to solve the problem by providing more eco-friendly and customized furniture options, helping people to find designs that perfectly match their needs and home styles to create unique and lasting spaces.

Q<sub>2</sub> who is your Target audience?

- people who care about their home and environment.
- Interior designers and architects.
- Small business owners like (cafes, restaurants, hotels) needs unique furniture.

Date 15/01/2025

Q3. What products or services will you offer?

- Custom furniture made from sustainable materials like (reclaimed wood, bamboo).
- DIY furniture kits with easy to follow assembly instructions.

Q4. What will set your marketplace apart?

- Working with local artisans for exclusive handcrafted designs.
- Carbon neutral delivery services and recyclable packaging.

Q. Data Schema:

1. identify the entities in your marketplace.

- products.
- Contact us or Help
- Orders
- Payment option
- Real-time tracking.
- User profiles.

2. Draw a Relationship

[Product] [Help]

- ID
- Issuing in payment
- Price
- " " tracking
- Stock



[Orders] → [Customer] → [Payment] → [Tracking]

- orderId
- CustomerId
- Paypal
- where is your product.
- ProductId
- Name
- Debit card
- Quantity
- Contact and Address
- Cash on delivery.
- Discount
- Email for promotion

## Day 2(Technical Problem)

```
D:\api\temp-3\paractice\template-2>npm run importdata
npm error Missing script: "importdata"
npm error
npm error Did you mean this?
npm error   npm run import-data # run the "import-data" package script
npm error
npm error To see a list of scripts, run:
npm error   npm run
npm error A complete log of this run can be found in: C:\Users\Abid\AppData\Local\npm-cache\_logs\2025-02-01T19_55_33_163Z-debug-0.log
```

## Day 3(Integrated Api And Migration-Data)

```
select npm run import-data
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/2219cafc285ec13a2ed3f88aa36cbea852a11735-305x375.png
Image uploaded successfully: image-2219cafc285ec13a2ed3f88aa36cbea852a11735-305x375-png
Imported product: Rustic Vase Set
Processing product: Bed
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/b6551d6252d30e222875fbc5d91c6edca324a1d-3518x3238.jpg
Image uploaded successfully: image-b6551d6252d30e222875fbc5d91c6edca324a1d-3518x3238-jpg
Imported product: Bed
Processing product: Wood Chair
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/2c7c233904e6c98e27a2156dce5f008b1554d25d-2831x3534.jpg
Image uploaded successfully: image-638c3039e0746f217acbdc325a2847939d92b08-2831x3534-jpg
Imported product: Wood Chair
Processing product: Retro Vibe
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/6e986db673401a7d253e8bdee86aceb4b71b4895-4144x4144.jpg
Image uploaded successfully: image-326af1f82195f9f3a6a9e8c882691aa81f2d0617-4144x4144-jpg
Imported product: Retro Vibe
Processing product: The Lucky Lamp
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/c10750f3c15f8807251977fe5e042b0a6cee55ab-305x375.png
Image uploaded successfully: image-c10750f3c15f8807251977fe5e042b0a6cee55ab-305x375-png
Imported product: The Lucky Lamp
Processing product: Pure Aura
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/db6d068faf67dab6145cadcd8e572944c6b9e935-3648x5472.jpg
Image uploaded successfully: image-db6d068faf67dab6145cadcd8e572944c6b9e935-3648x5472-jpg
Imported product: Pure Aura
Processing product: Zen Table
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/73e36a9afc074b751d9f2693d2b251ba12b36276-3433x3064.jpg
Image uploaded successfully: image-2e9468c5de6d27c850d6c6ba74ee16bcbf200d77a-3433x3064-jpg
Imported product: Zen Table
Processing product: Tropical Vibe
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/9dd7acc17365cd290757246a27b64a3d817che13-5501x3095.jpg
```



## Day 4(Fetch Data, Dynamic Routing)

Complexity is 5 Everything is cool!

```
export const getProductBySlug = async (slug: string) => {  
  const PRODUCT_BY_SLUG_QUERY = defineQuery(  
    `*[_type == 'product' && slug.current == $slug] | order(name asc) [0]`  
  );  
  try {  
    const product = await sanityFetch({  
      query: PRODUCT_BY_SLUG_QUERY,  
      params: {  
        slug,  
      },  
    });  
    return product?.data || null;  
  } catch (error) {  
    console.error("Error fetching product by Slug:", error);  
  }  
};
```

Complexity is 39 Bloody hell...












```
const SingleProductPage = async ({  
  params,  
}): {  
  params: Promise<{ slug: string }>;  
} => {  
  const { slug } = await params;  
  const product = await getProductBySlug(slug);  
  if (!product) {  
    return notFound();  
  }  
  
  return (  
    <Container className="py-10 flex flex-col md:flex-row gap-10">  
      {product?.image} && <ImageView image={product?.image} />  
      <div className="w-full md:w-1/2 flex flex-col gap-5">  
        <div>  
          <h2 className="text-3xl md:text-4xl font-bold mb-2">  
            {product?.name}  
          </h2>  
          <PriceView  
            price={product?.price}  
            discount={product?.discount}  
            className="text-lg font-bold"  
          />  
        </div>  
        {product?.quantity} && (  
          <p className="bg-green-100 w-24 text-center text-green-600 text-sm py-2.5 font-semibold rounded-lg">  
            In Stock  
          </p>  
        )  
      </div>  
    </Container>  
  );  
};
```

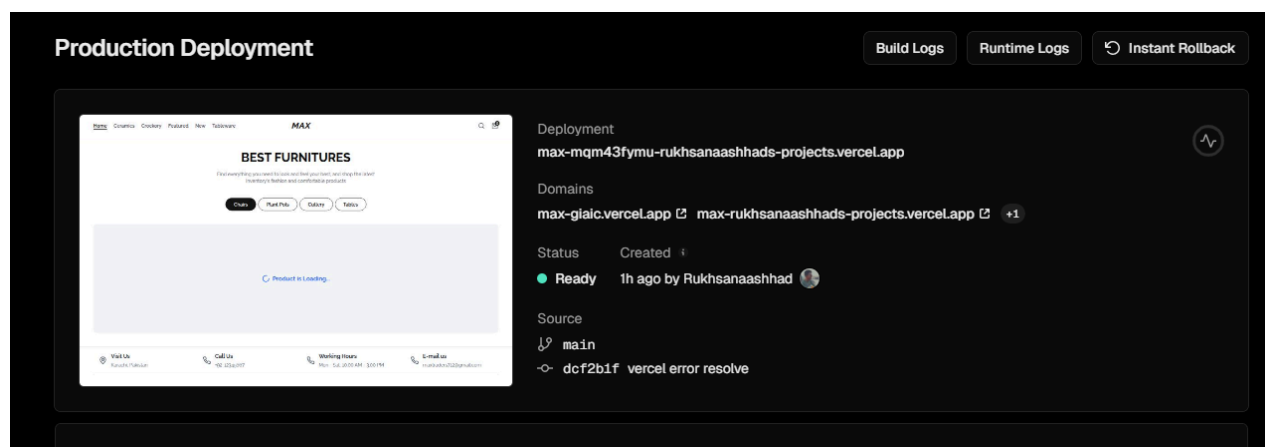
## Day 5(Error Handling)

```
page.tsx 3  TS queries.ts
app > (client) > category > [slug] > page.tsx > CategoryPage
1  import CategoryProducts from '@components/CategoryProducts';
2  import Container from '@components/Container'
3  import Title from '@components/Title';
4  import { getAllCategories } from '@sanity/helpers/queries';
5  import React from 'react'
6
7  Complexity is 5 Everything is cool!
8  const CategoryPage = async ({params}:{params:Promise<{slug: string}>});
9  }) => {
10     const { slug } = await params;
11     const categories = await getAllCategories();
12     return (
13       <Container className="py-10">
14         <Title className="text-xl">
15           Products by Category
16           { /*<span className="font-bold text-green-600 capitalize tracking-wide">{slug && slug }</span> */}
17         </Title>
18         <CategoryProducts categories={categories} slugs={slug} />
19       </Container>
20     )
21   }
22 }
23
24 export default CategoryPage
25
```

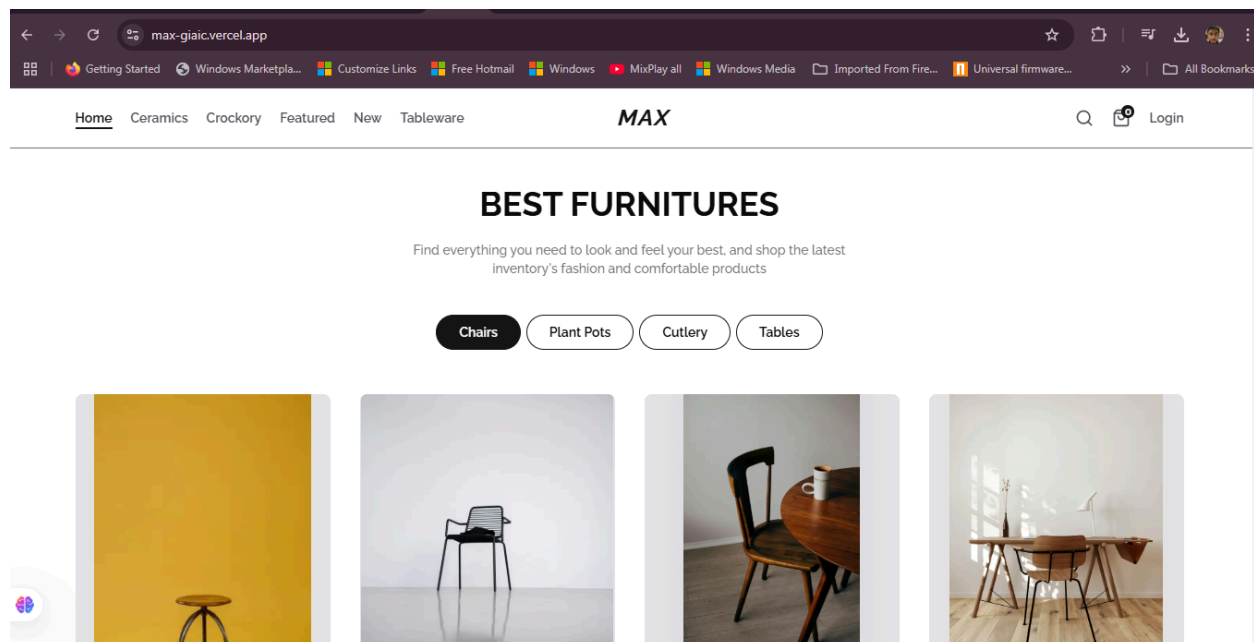
## Day 6(Deployment Stage)

```
17:17:00.024 Failed to compile.
17:17:00.024
17:17:00.025 ./components/ImageView.tsx
17:17:00.025 10:10 Error: The `{}` ("empty object") type allows
any non-nullish value, including literals like `0`
and `""`.
17:17:00.025 - If that's what you want, disable this lint rule w
ith an inline comment or configure the 'allowObject
Types' rule option.
17:17:00.025 - If you want a type meaning "any object", you prob
ably want `object` instead.
17:17:00.025 - If you want a type meaning "any value", you proba
bly want `unknown` instead. @typescript-eslint/no-
empty-object-type
17:17:00.025
17:17:00.025 ./components/ProductGrid.tsx
17:17:00.025 35:7 Warning: React Hook useEffect has missing dep
```

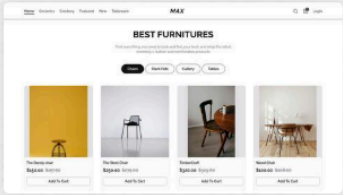
Time	Status	Host	Request	Messages
FEB 02 00:59:13.43	GET 200	max-giaic.ver..	 /product/the-lucky-lamp	
FEB 02 00:59:13.42	GET 200	max-giaic.ver..	 /product/rustic-vase-set	
FEB 02 00:59:10.25	GET 200	max-giaic.ver..	 /product/marble-ease	
FEB 02 00:59:08.65	GET 200	max-giaic.ver..	 /product/reflective-haven	
FEB 02 00:59:08.65	GET 200	max-giaic.ver..	 /product/modern-serenity	
FEB 02 00:59:04.70	GET 200	max-giaic.ver..	 /product/bold-nest	
FEB 02 00:59:04.70	GET 200	max-giaic.ver..	 /product/bed	
FEB 02 00:59:01.63	GET 200	max-giaic.ver..	 /product/sunny-chic	
FEB 02 00:59:01.63	GET 200	max-giaic.ver..	 /product/sleek-living	
FEB 02 00:59:00.90	GET 200	max-giaic.ver..	 /product/the-poplar-suede-sofa	
FEB 02 00:58:57.82	GET 200	max-giaic.ver..	 /category/tableware	





## Day 7(Live)




# Website Performance



## Latest Performance Report for: <https://max-giaic.vercel.app/>

Report generated: Sat, Feb 1, 2025 11:43 AM -0800  
Test Server Location:  Vancouver, Canada  
Using:  Chrome 117.0.0.0, Lighthouse 11.0.0

### GTmetrix Grade ?

	Performance ? 89%	Structure ? 94%
---	----------------------	--------------------

### Web Vitals ?

LCP ? 1.8s	TBT ? 21ms	CLS ? 0.01
---------------	---------------	---------------