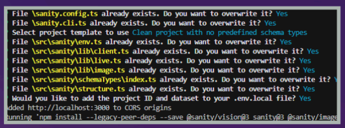
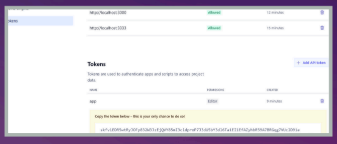
# Here’s a step-by-step outline of your work

. **E-commerce UI/UX Design**
Design and develop the user interface and user experience for the e-commerce platform.



```
File \sanity.config.ts already exists. Do you want to overwrite it? Yes
File \sanity.cli.ts already exists. Do you want to overwrite it? Yes
Select project template to use Clean project with no predefined schema types
File \src\sanity\env.ts already exists. Do you want to overwrite it? Yes
File \src\sanity\lib\client.ts already exists. Do you want to overwrite it? Yes
File \src\sanity\lib\live.ts already exists. Do you want to overwrite it? Yes
File \src\sanity\lib\image.ts already exists. Do you want to overwrite it? Yes
File \src\sanity\schemaTypes\index.ts already exists. Do you want to overwrite it? Y
File \src\sanity\structure.ts already exists. Do you want to overwrite it? Yes
Would you like to add the project ID and dataset to your .env.local file? Yes
dded http://localhost:3000 to CORS origins
unning 'npm install --legacy-peer-deps --save @sanity/vision@3 sanity@3 @sanity/image
```

. **Create Sanity Project**
Set up a new Sanity project to use as a content management system.



okens

http://localhost:3000 Allowed 12 minutes

http://localhost:3333 Allowed 15 minutes

**Tokens**

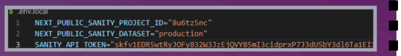
Tokens are used to authenticate apps and scripts to access project data.

| NAME | PERMISSIONS | CREATED |
|---|---|---|
| app | Editor | 9 minutes |

Copy the token below – this is your only chance to do so!

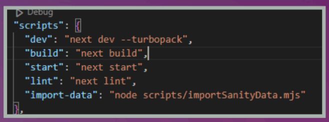
skfv1EDR5wtRyJOFy832W3JzEjQVY85mI3cidprxP7J3dU5bY3dl6Ta1EI1EfAZyhbR59A7BRGqg7VUclD91e

. **Embed Sanity**
Integrate Sanity into your e-commerce project to manage content dynamically.



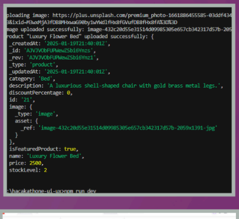
```
$ .env.local
NEXT_PUBLIC_SANITY_PROJECT_ID="8u6tz5nc"
NEXT_PUBLIC_SANITY_DATASET="production"
SANITY_API_TOKEN="skfv1EDR5wtRyJOFy832W3JzEjQVY85mI3cidprxP7J3dU5bY3dl6Ta1EI
```

. **Schema of Product**
Define the schema for products in Sanity, including fields like name, description, price, and images.



```
▷ Debug
"scripts": {
  "dev": "next dev --turbopack",
  "build": "next build",
  "start": "next start",
  "lint": "next lint",
  "import-data": "node scripts/importSanityData.mjs"
},
```

. **Configure .env File**
Add your Sanity project ID and token to the .env file for secure access.



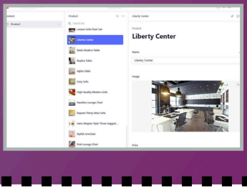
```
ploading image: https://plus.unsplash.com/premium_photo-1661886455585-03ddf434
3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D
mage uploaded successfully: image-432c20d55e31514d09985305e657cb342317d57b-205
roduct "Luxury Flower Bed" uploaded successfully: {
 _createdAt: '2025-01-19T21:40:01Z',
 _id: 'AJVJVObFUPWewZSbi6Ynzs',
 _rev: 'AJVJVObFUPWewZSbi6Ynz1',
 _type: 'product',
 _updatedAt: '2025-01-19T21:40:01Z',
 category: 'Bed',
 description: 'A luxurious shell-shaped chair with gold brass metal legs.',
 discountPercentage: 0,
 id: '21',
 image: {
   _type: 'image',
   asset: {
     _ref: 'image-432c20d55e31514d09985305e657cb342317d57b-2059x1391-jpg'
   }
 },
 isFeaturedProduct: true,
 name: 'Luxury Flower Bed',
 price: 2500,
 stockLevel: 2


:\hacakathone-ui-ux>npm run dev
```

. **Create m.js for Fetching Data**
Develop a file m.js to handle fetching data from Sanity's API.

. **Update package.json**
Modify package.json to include and utilize the m.js file for data operations.

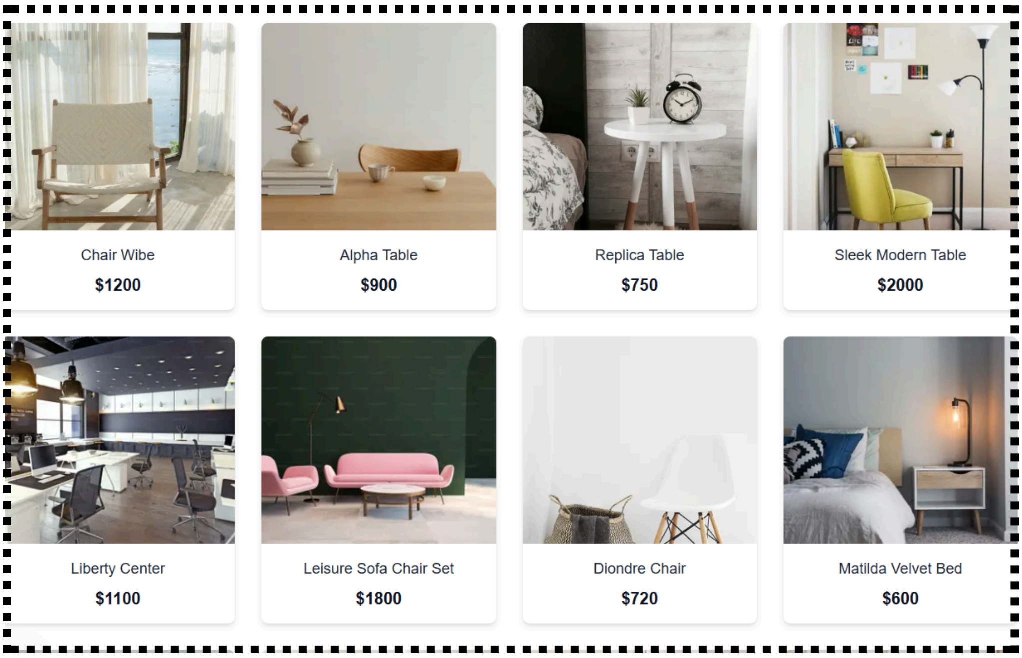
. **Insert Data into Sanity**
Populate Sanity with product data, making it available for fetching through the API.



**Liberty Center**

Name: Liberty Center

## Data fetching on display from API:



Chair Wibe
**$1200**

Alpha Table
**$900**

Replica Table
**$750**

Sleek Modern Table
**$2000**

Liberty Center
**$1100**

Leisure Sofa Chair Set
**$1800**

Diondre Chair
**$720**

Matilda Velvet Bed
**$600**

# My Day 3 Assigment:

**API Integration**: The project involves seamless integration of APIs to facilitate data exchange between the backend and frontend. This ensures dynamic content delivery and a responsive user experience.

**Updated Schema into Sanity**: The schema in Sanity has been updated to align with the latest data requirements. This ensures that the content structure is well-organized and scalable, accommodating future enhancements and new features.

**Migration Steps and Tools Used**: The migration process involved steps like data extraction, transformation, and loading into the new system. Tools such as data migration frameworks and scripts were utilized to ensure a smooth transition with minimal downtime and data integrity.

**API Calls**: Various API calls have been implemented to fetch and update data in real-time. These include endpoints for retrieving product details, user information, and managing orders, enhancing the functionality and interactivity of the application.

**Data Successfully Displayed in Frontend**: Post-integration, the data fetched via APIs is displayed successfully on the frontend. This achievement marks a critical milestone in the project, showcasing a fully functional and user-friendly interface that interacts seamlessly with the backend.

# Conclusion:

The successful implementation of API integration, the update of the schema in Sanity, and the effective execution of migration steps have significantly contributed to the project's success. By employing reliable tools and precise API calls, data is seamlessly managed and displayed on the frontend, resulting in an enhanced user experience. This robust and scalable system is well-prepared to accommodate future enhancements, ensuring long-term adaptability and growth. The project exemplifies the importance of meticulous planning and execution in delivering a dynamic and user-friendly platform.