

"Day 2-Technical Foundation - [Evion E-commerce]"

1. Technical Plan:

- **Frontend:** Next.js, Tailwind CSS, shadcn/ui
- **Backend:** Sanity CMS (for content management), API for product data and user management.
- **Database:** Sanity for CMS content or another database (if needed) for additional user data.
- **Payment Gateway:** Stripe, PayPal, or another provider.
- **Hosting:** Vercel (for Next.js deployment), Sanity for CMS hosting.
- **Version Control:** GitHub for code management and collaboration.
- **Authentication:** JWT tokens for secure login sessions, using Next.js API routes for user registration/login.

2. Workflows:

User Registration/Authentication Workflow:

- Users sign up or log in using email/password or third-party OAuth.

Product Search and Filter Workflow:

- Users can search and filter products by type (furniture, decor), price range, and availability.

Product Details and Cart Workflow:

- Users click on a product for more details, add it to the cart, and proceed to checkout.

Checkout and Payment Workflow:

- After confirming the cart, users proceed to checkout and make a payment via a gateway (e.g., Stripe).

Admin Workflow:

- Admins can manage products, view orders, and update inventory via the CMS dashboard.

3. API Requirements:

1. Endpoint: /products

Method: GET

Description: Fetch all available products from Sanity.

Response Example:

```
[  
  
  {  
  
    "id": 1,  
  
    "name": "Product A",  
  
    "price": 100,  
  
    "stock": 20,  
  
    "image": "image_url"  
  },  
  
  {  
  
    "id": 2,  
  
    "name": "Product B",  
  
    "price": 150,  
  
    "stock": 10,  
  
    "image": "image_url"  
  }  
]
```

]

2. Endpoint: /orders

Method: POST

Description: Create a new order in Sanity.

Payload:

```
{  
  
  "customerInfo": {  
  
    "name": "John Doe",  
  
    "email": "john@example.com",  
  
    "address": "123 Main St"  
  
  },  
  
  "products": [  
  
    {  
  
      "productId": 1,  
  
      "quantity": 2  
  
    },  
  
    {  
  
      "productId": 2,  
  
      "quantity": 1  
  
    }  
  
  ],  
  
  "paymentStatus": "Pending"  
  
}
```

3. Endpoint: /shipment

Method: GET

Description: Track order status via third-party API.

Response Example:

```
{  
  
  "shipmentId": 456,  
  
  "orderId": 123,  
  
  "status": "In Transit",  
  
  "expectedDeliveryDate": "2025-02-05"  
}
```

4. Sanity Schema:

Product Schema:

```
export default {  
  
  name: 'product',  
  
  title: 'Product',  
  
  type: 'document',  
  
  fields: [  
  
    { name: 'name', type: 'string' },  
  
    { name: 'description', type: 'text' },  
  
    { name: 'price', type: 'number' },  
  
    { name: 'image', type: 'image' },  
  
    { name: 'type', type: 'string', options: { list: ['furniture', 'decor'] } },  
  
    { name: 'available', type: 'boolean' },  
  
  ]  
}
```

```
};
```

User Schema:

```
export default {  
  
  name: 'user',  
  
  title: 'User',  
  
  type: 'document',  
  
  fields: [  
  
    { name: 'name', type: 'string' },  
  
    { name: 'email', type: 'string' },  
  
    { name: 'password', type: 'string' },  
  
    { name: 'orders', type: 'array', of: [{ type: 'reference', to: [{ type: 'order' }] }] },  
  
  ]  
  
};
```

Order Schema:

```
export default {  
  
  name: 'order',  
  
  title: 'Order',  
  
  type: 'document',  
  
  fields: [  
  
    { name: 'user', type: 'reference', to: [{ type: 'user' }] },  
  
    { name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }] },  
  
    { name: 'totalAmount', type: 'number' },  
  
    { name: 'status', type: 'string', options: { list: ['pending', 'completed', 'shipped'] } },  
  
  ]  
  
};
```

```
{ name: 'shippingAddress', type: 'string' },  
]  
};
```

