

Marketplace Technical Foundation

- FURNINEST

FATIMA AKIF | roll no: 00312734

Marketplace Technical Foundation - FurniNest

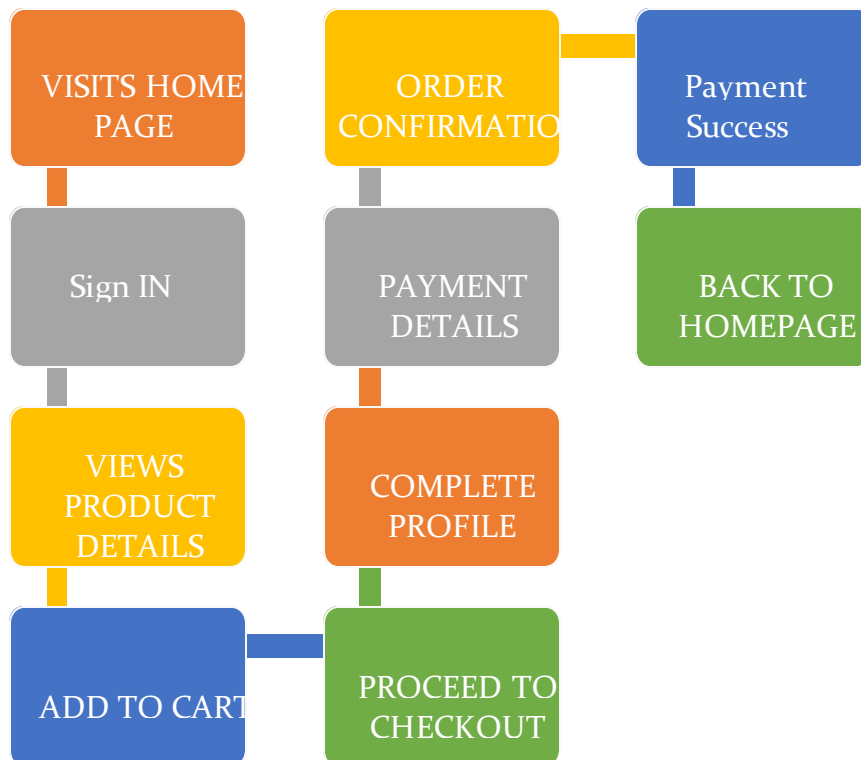
FurniNest is a comprehensive e-commerce marketplace specializing in furniture sales. It provides a seamless platform for customers to explore, purchase, and manage furniture products with ease. The marketplace integrates essential technologies to ensure a secure, scalable, and efficient online shopping experience.

Core Technologies

1. **Sanity** – A powerful content management system (CMS) that enables dynamic and structured product management.
2. **Stripe** – A robust payment gateway that facilitates secure and seamless transactions.
3. **Clerk** – A user authentication and management system that ensures secure access and personalized user experiences.

FurniNest is designed to deliver a smooth and user-friendly shopping journey, leveraging modern web technologies to enhance performance, security, and usability.

1. System Architecture



Overview

The marketplace will be a **Next.js** application with a backend powered by **Sanity CMS** for content management. The system will follow a **headless CMS approach**, ensuring flexibility, scalability, and modularity.

Architecture Components

- **Frontend:** Next.js with Tailwind CSS for a responsive UI.
- **Backend:** Sanity CMS for managing product data.
- **Database:** Sanity's GROQ-powered database.
- **Authentication:** Firebase Authentication for user sign-up and login.
- **Payment Gateway:** Stripe for secure transactions.
- **APIs:** RESTful API and GraphQL endpoints for data retrieval.
- **Deployment:** Vercel (Frontend) and Sanity's hosted backend.

System Flow

1. User visits the marketplace.
2. Fetch products from Sanity CMS.
3. Authentication using Clerk.
4. Users can add products to their cart.
5. Checkout process via Stripe.
6. Orders stored and managed through Sanity CMS.

2. API Requirements

Authentication API

- **POST /api/auth/signup** – Register a new user.
- **POST /api/auth/login** – Authenticate user.
- **POST /api/auth/logout** – Log out user.

Products API

- **GET /api/products** – Fetch all products.
- **GET /api/products/{id}** – Fetch a single product.
- **POST /api/products** – Add a new product (Admin access required).
- **PUT /api/products/{id}** – Update product details (Admin only).
- **DELETE /api/products/{id}** – Remove a product (Admin only).

Cart API

- **POST** `/api/cart/add` – Add item to cart.
- **GET** `/api/cart` – Retrieve cart items.
- **DELETE** `/api/cart/remove/{id}` – Remove item from cart.

.

3. Sanity CMS Schema

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'title',
      title: 'Title',
      type: 'string'
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text'
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number'
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image'
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string'
    },
    {
      name: 'stock',
      title: 'Stock Quantity',
      type: 'number'
    }
  ]
};
```

This document sets up a robust foundation for the marketplace, covering essential architecture, API endpoints, and a CMS schema. Let me know if you need any modifications!