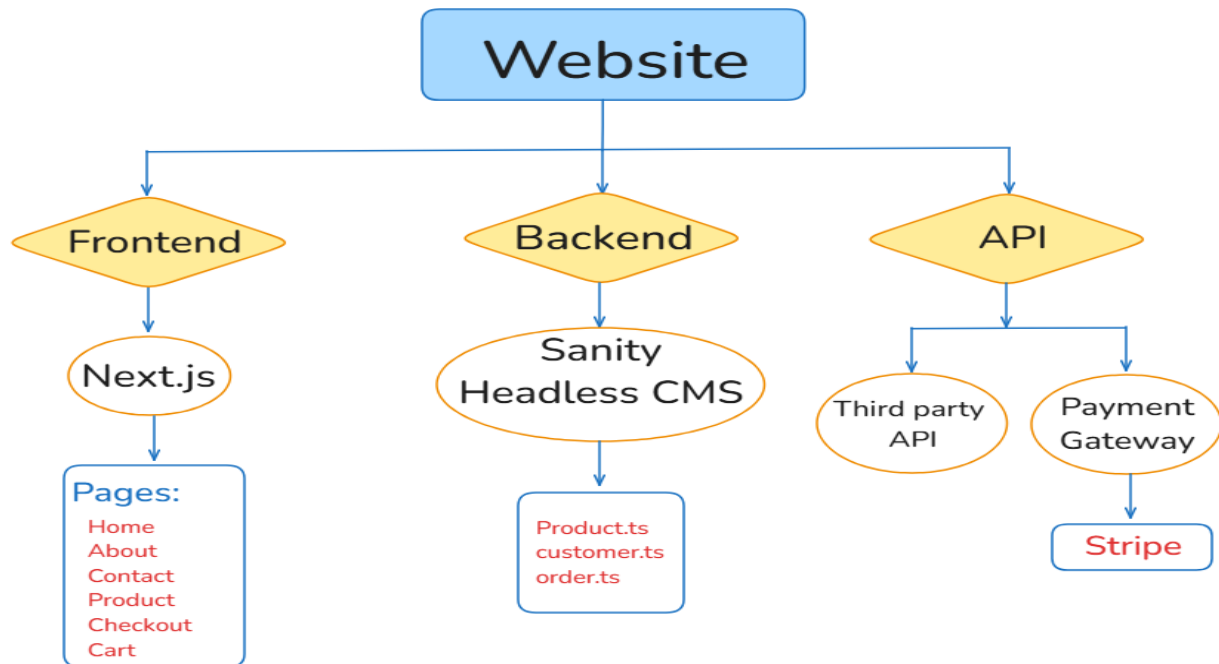


Marketplace Technical Foundation – [E-Commerce]

System Architecture:



Key Features:

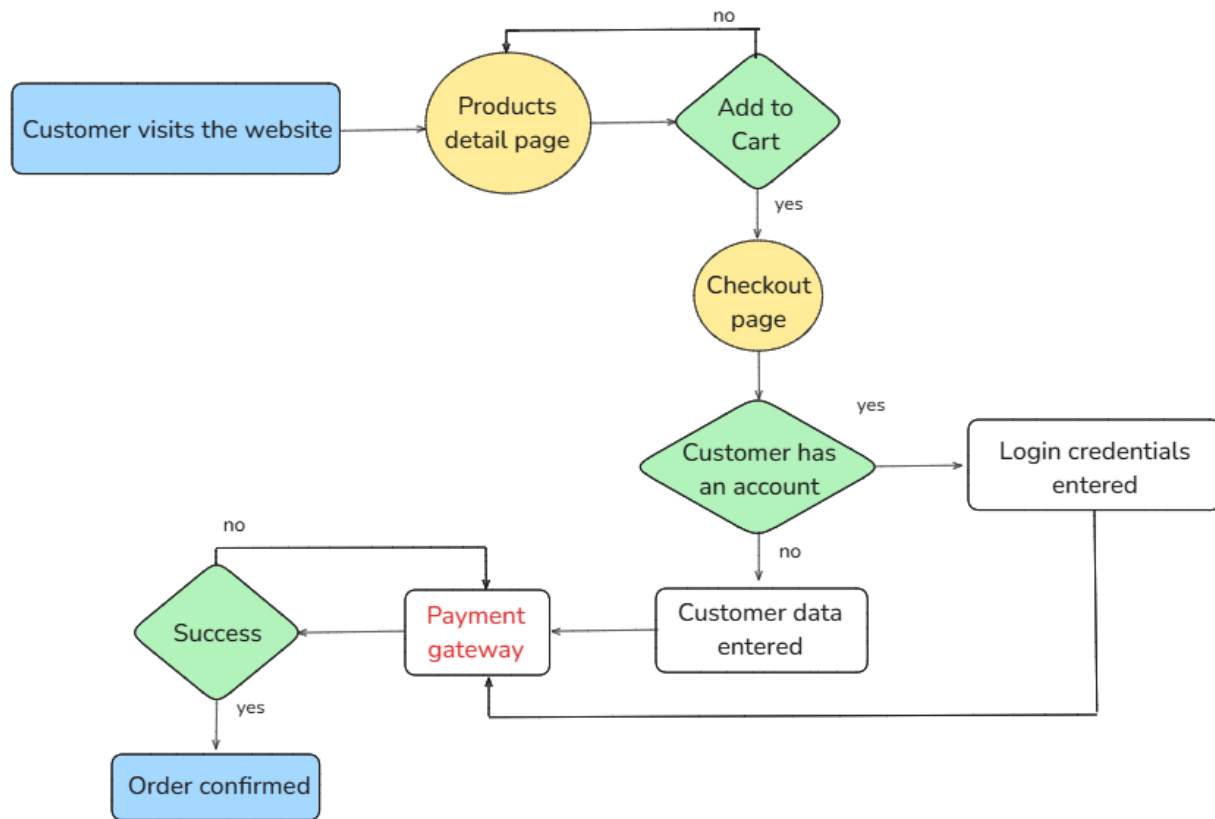
Frontend

- Built with **Next.js**, **TypeScript**, **Tailwind CSS**, and custom CSS.
- Fully responsive for mobile, tablet, and desktop users.
- Pages: **Home**, **About**, **Products**, **Contact**, **Cart**, **Checkout**.
- Modular components for consistency and reusability.

Backend (Sanity CMS)

- **Sanity CMS** manages:
 - Product data, customer details, order records, recommendations, shipment, and payment info.
- **Third-party APIs**: Fetch product data, process payments, manage shipping, and provide recommendations.
- User data stored in Sanity CMS.

Workflow:



API Endpoints for E-Commerce:

Customer Management:

```
customer: {  
  create: "POST /api/customer",  
  getAll: "GET /api/customer",  
  getOne: "GET /api/customer/:id",  
  update: "PUT /api/customer/:id",  
  delete: "DELETE /api/customer/:id",  
},
```

Product Management:

```
product: {  
  create: "POST /api/product",  
  getAll: "GET /api/product",  
  getOne: "GET /api/product/:id",  
  update: "PUT /api/product/:id",  
  delete: "DELETE /api/product/:id",  
},
```

Category Management:

```
category: {  
  create: "POST /api/category",  
  getAll: "GET /api/category",  
  getOne: "GET /api/category/:id",  
  update: "PUT /api/category/:id",  
  delete: "DELETE /api/category/:id",  
},
```

Order Management:

```
order: {  
  create: "POST /api/order",  
  getAll: "GET /api/order",  
  getOne: "GET /api/order/:id",  
  update: "PUT /api/order/:id",  
  delete: "DELETE /api/order/:id",  
},
```

Payment Management:

```
payment: {  
  initiate: "POST /api/payment/initiate",  
  confirm: "POST /api/payment/confirm",  
  refund: "POST /api/payment/refund",  
  status: "GET /api/payment/status/:id",  
},
```

Shipment Management:

```
shipment: {  
  create: "POST /api/shipment",  
  getAll: "GET /api/shipment",  
  getOne: "GET /api/shipment/:id",  
  update: "PUT /api/shipment/:id",  
  delete: "DELETE /api/shipment/:id",  
},
```

Recommendations Management:

```
recommendation: {  
  getAll: "GET /api/recommendation",  
  getUser: "GET /api/recommendation/user/:id",  
  create: "POST /api/recommendation",  
  update: "PUT /api/recommendation/:id",  
  delete: "DELETE /api/recommendation/:id",  
},
```