

# Day 4 – Dynamic Frontend Components – Home haven

## Implementation Steps

### Setup Phase

- Connected Next.js project with Sanity CMS
- Set up required dependencies and configurations
- Created component folder structure

### Core Components Built

#### 1. Product Listing Component

- Implemented grid layout with responsive design
- Added dynamic data fetching from Sanity CMS

#### 2. Product Detail Pages

- Created dynamic routing for individual products
- Added image gallery and product information display

#### 3. Add to cart & wish list page

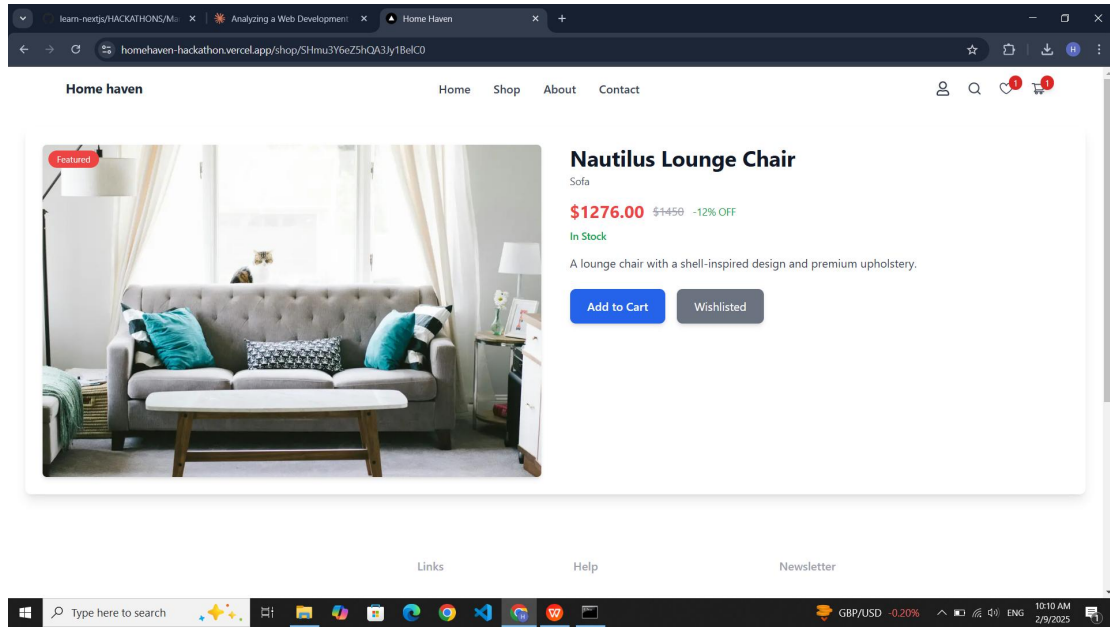
- Created add to cart page through context API
- Created wish list page through context API

#### 4. Confirmation Email After order placing

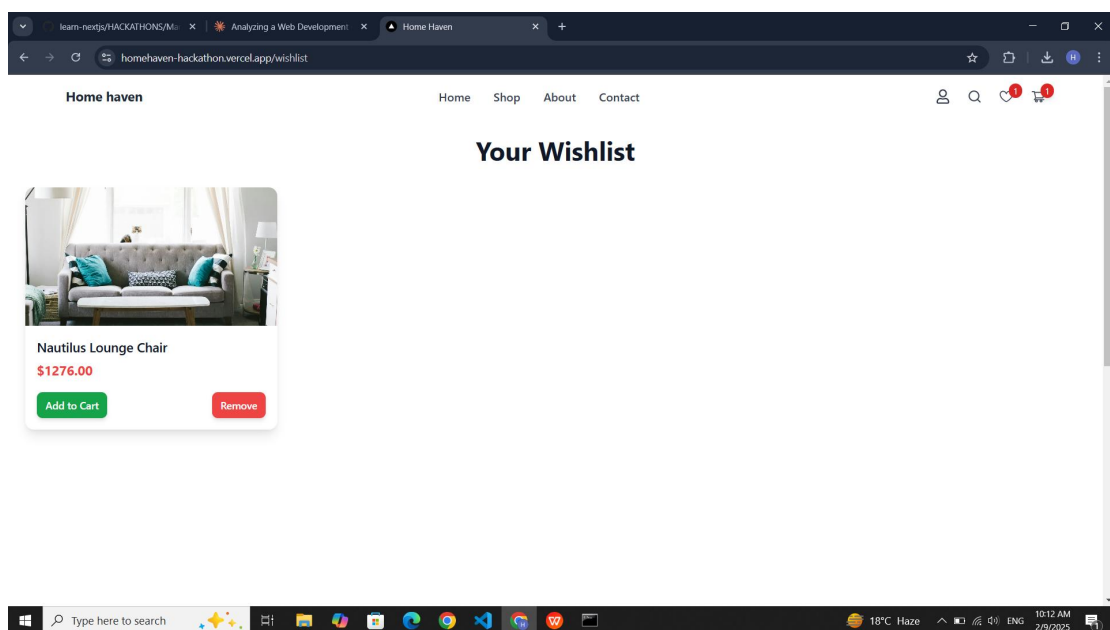
- Using resend API for confirmation email after order

# Screenshots of some components

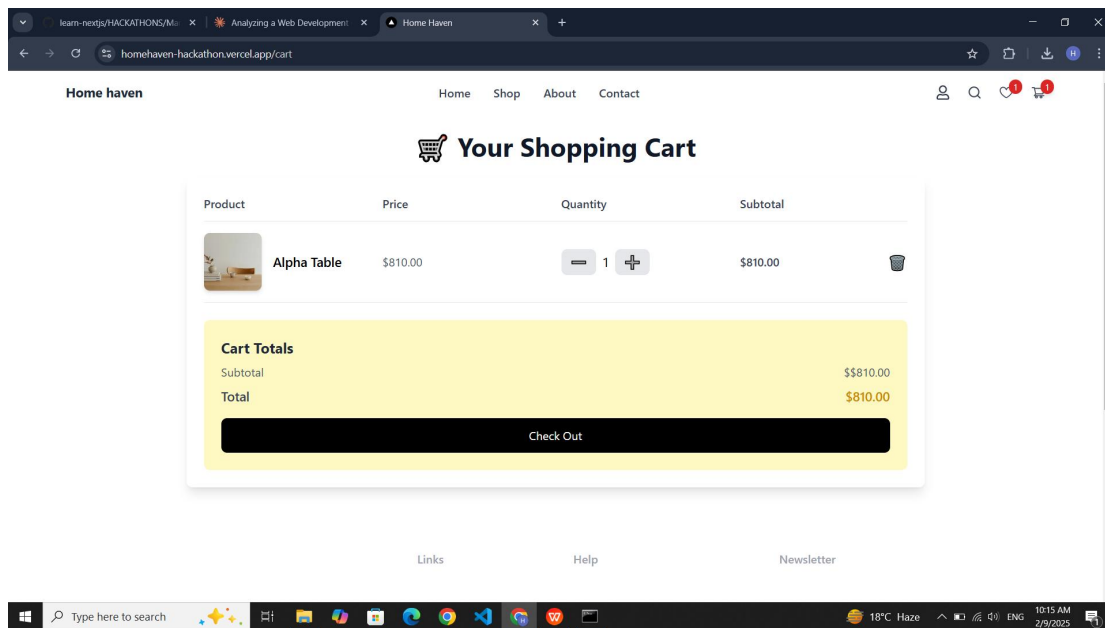
## 1. Dynamic routes



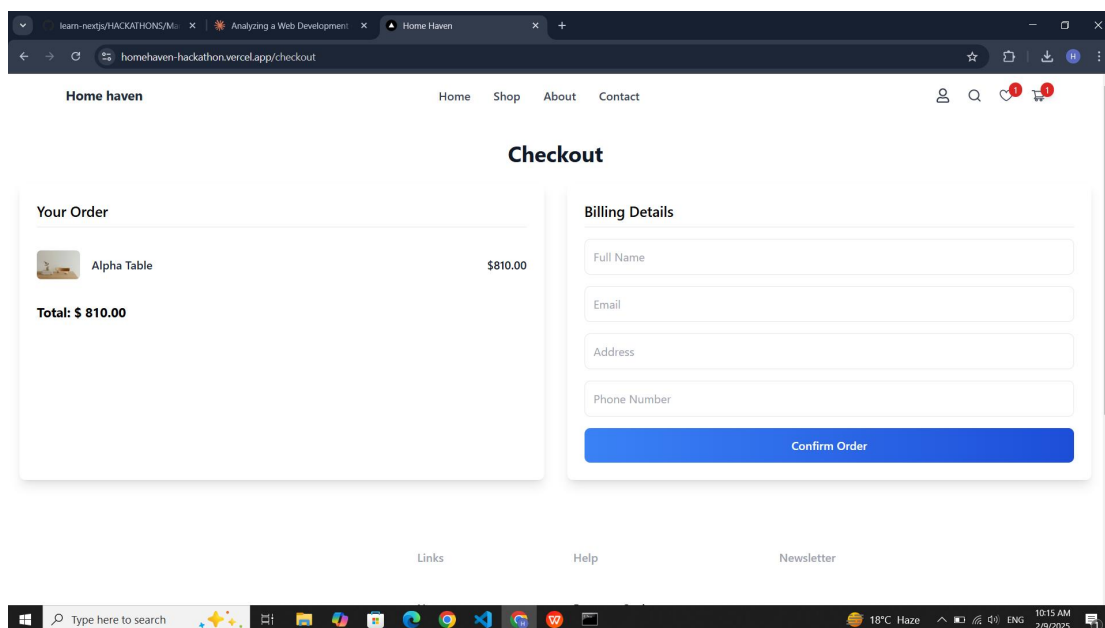
## 2. Whish list



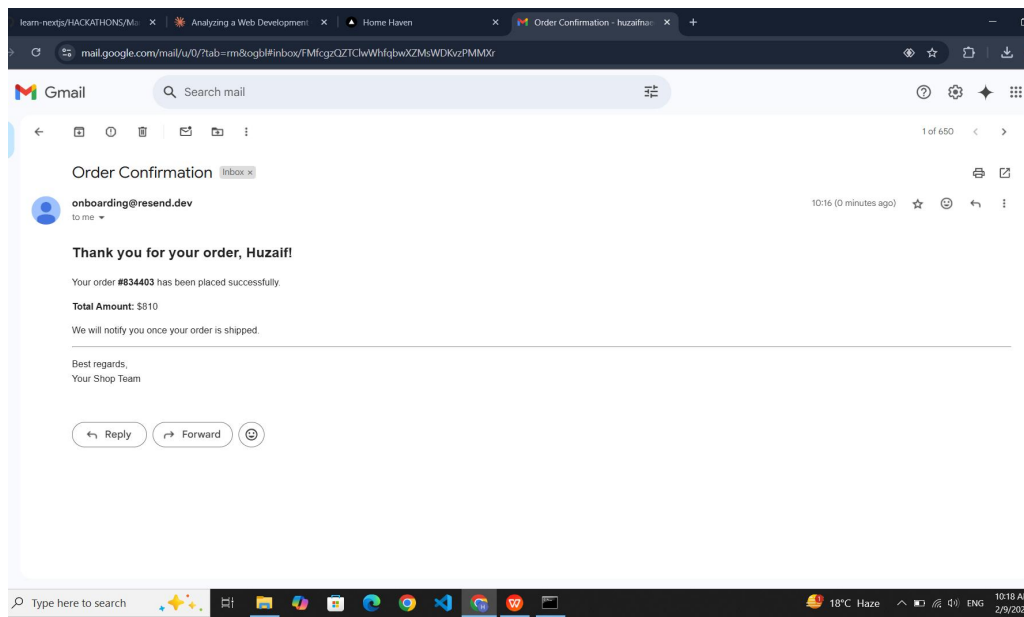
### 3. Add to cart



### 4. Checkout Page



## 4. Confirmation Email



## Challenges and Solutions

### 1. State Management

- Challenge: Managing cart state across components
- Solution: Used React Context for global state management

### 2. Responsive Design

- Challenge: Layout issues on mobile devices
- Solution: Implemented mobile-first approach with CSS Grid

## Best Practices Applied

### 1. Code Organization

- Used modular component structure
- Implemented TypeScript for type safety

### 2. Performance

- Optimized images using Next.js Image component
- Implemented lazy loading for components

### 3. User Experience

- Added error handling
- Included loading indicators
- Ensured responsive design across devices

## Future Improvements

1. Add user authentication
2. Implement advanced search features
3. Add more payment options
4. Enhance admin dashboard functionality

This project successfully implements the core marketplace features while maintaining good performance and user experience standards.