

Technical Requirements

Front End Requirements

- Next.js with Typescript, Schdcn/ui, React-Icons, Tailwinds .
- Important Pages such as {Name, Product listening, Product Details, Cart, Checkout, Order Confirmation, Shipment Tracking, }.
- User Friendly Interface.
- Responsive design for both mobile and Laptops

Back End Requirements

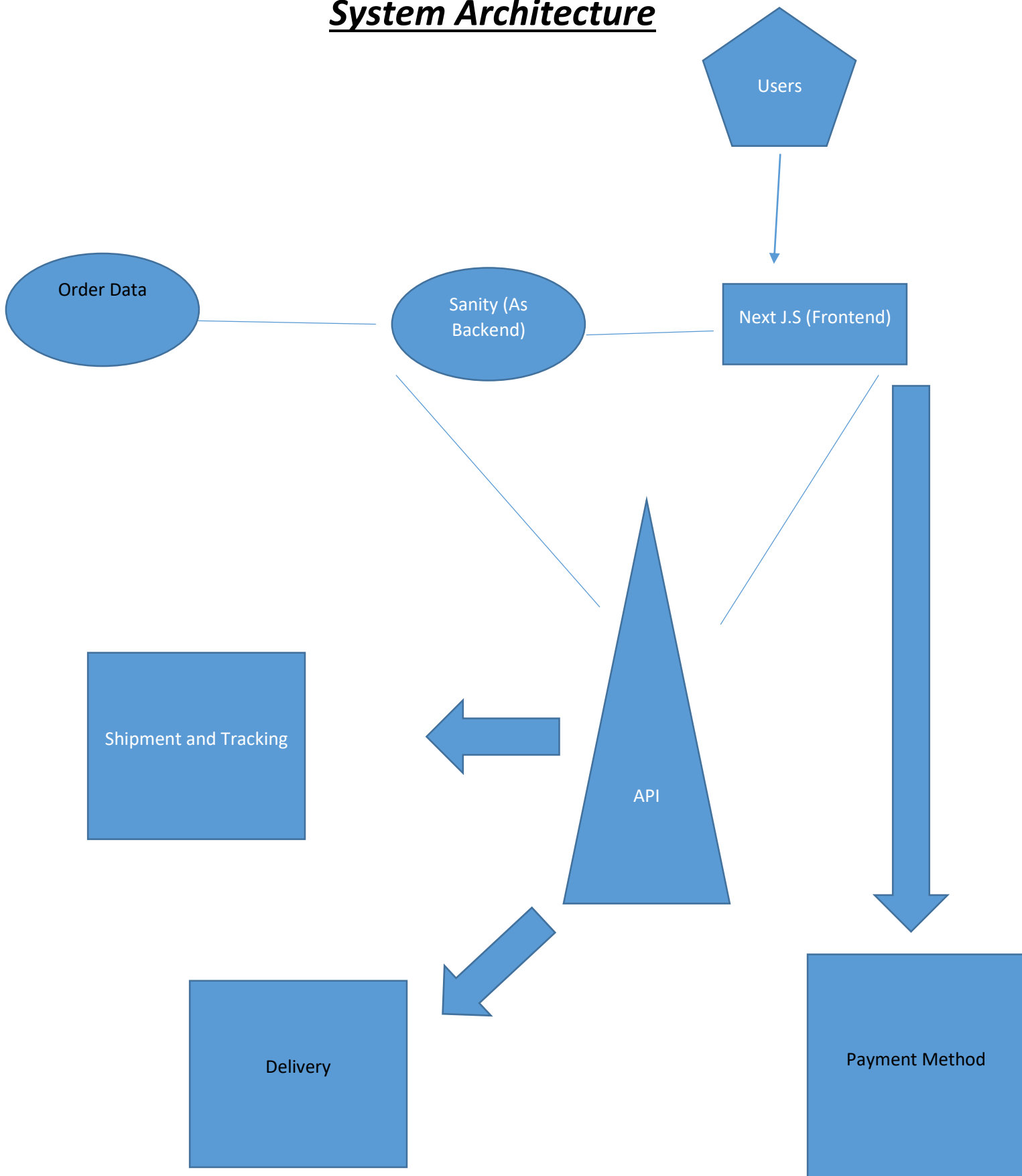
Sanity CMS required for fetching Data

- Product Data
- Customer Data
- Order Data

Third-Party APIs:

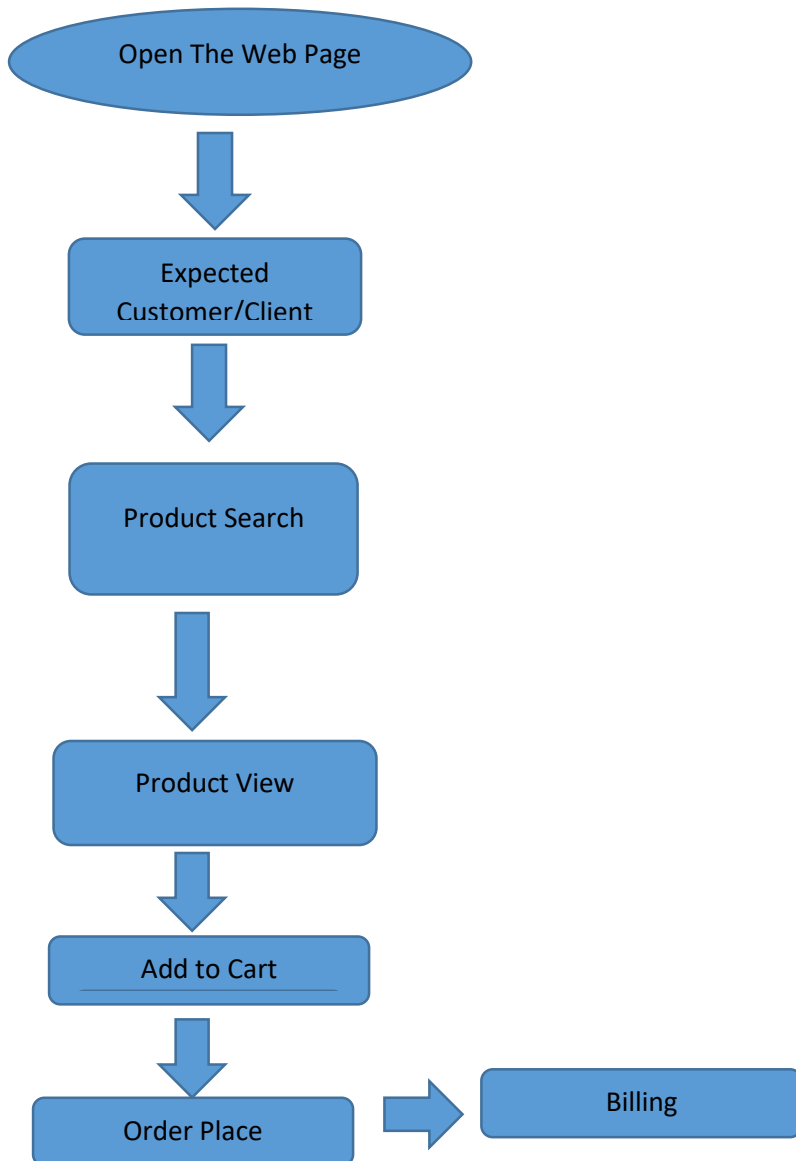
- Payment integration : using stripe
- Shipment Tracking: using ship engine, shippoar integration.

System Architecture

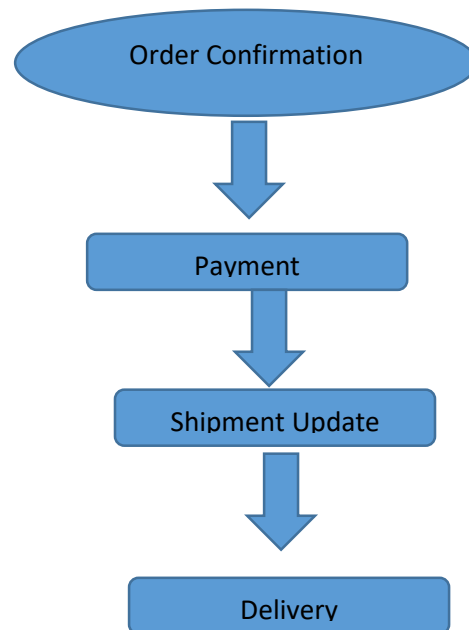


Flow Chart of the E-Commerce Website

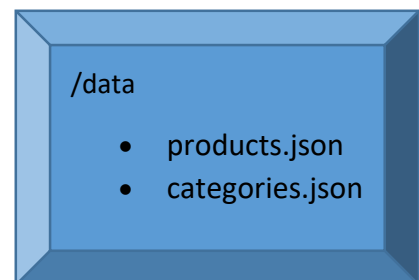
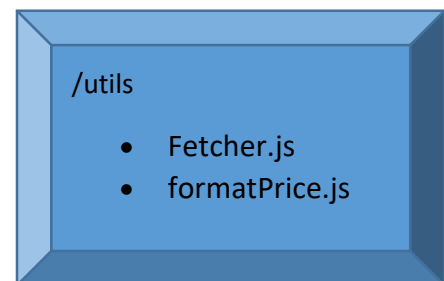
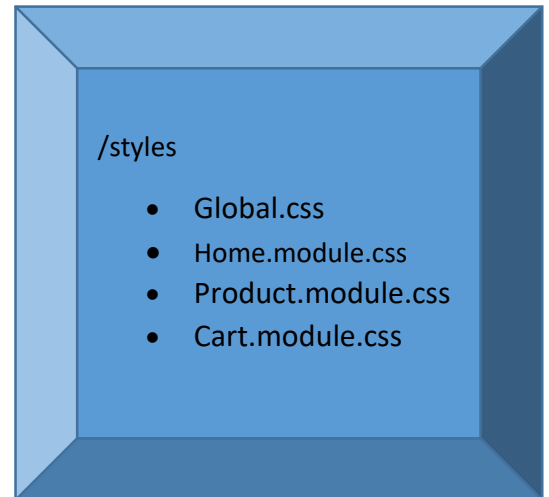
Customer Side



Admin Side



Project Structure



Sanity Schema

```
export default {
```

```
name: 'product',
type: 'document',
title: 'Product',
fields: [
  {
    name: 'name',
    type: 'string',
    title: 'Name',
  },
  {
    name: 'image',
    type: 'image',
    title: 'Image',
    options: {
      hotspot: true, // Enable image cropping
    },
  },
  {
    name: 'price',
    type: 'number',
    title: 'Price',
  },
  {
    name: 'description',
    type: 'text',
    title: 'Description',
  },
  {
    name: 'discountPercentage',
    type: 'number',
    title: 'Discount Percentage',
  },
  {
    name: 'isFeaturedProduct',
    type: 'boolean',
    title: 'Featured Product',
  },
  {
    name: 'stockLevel',
    type: 'number',
    title: 'Stock Level',
  },
  {
    name: 'category',
    type: 'string',
```

```
        title: 'Category',  
    },  
],  
};
```

Product Data (Product. JSON)

```
[  
{  
  "id":1,  
  "Name":"CoffeeTable",  
  "Price":500.00,  
  "description":"Modern and Stylish Table",  
  "image": "/images/table.jpg",  
  "category":"Resturants/ Offices",  
  "Stock":20  
},  
{  
  "id":2  
  "Name":" Dining Table and Chair",  
  "Price":800,  
  "description": "Luxury Table and Chairs",  
  "image": "/images/dining-table.jpg",
```

```
    "category": "Dining Room",  
    "Stock": 5  
  }  
]
```

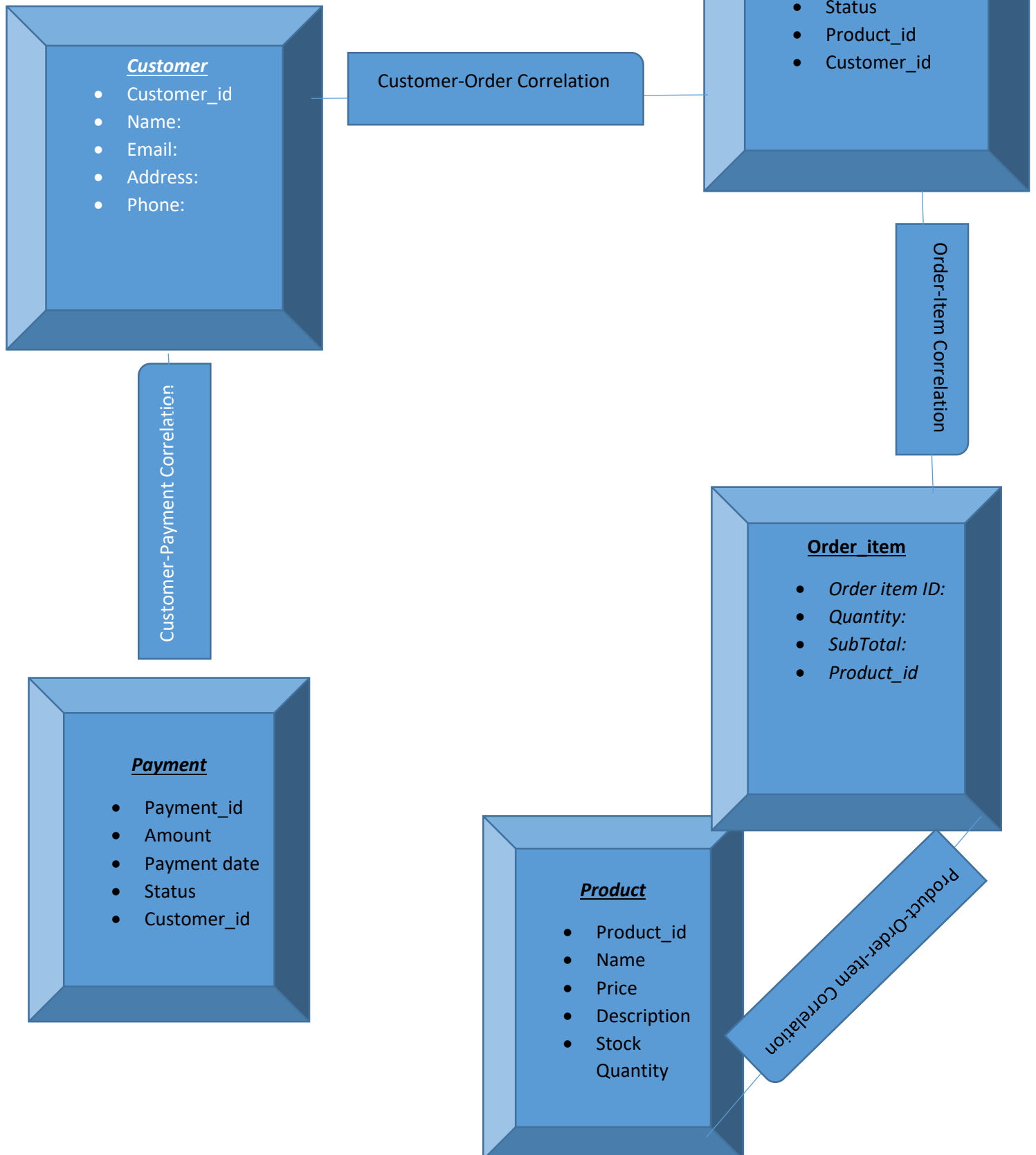
Category Data (Categories.json)

```
[  
  {  
    "id":1,  
    "name": "Offices/Resturants",  
    "slug": "resturants",  
  },  
  {  
    "Id":2,  
    "name": "Dining Room",  
    "slug": "dining-room",  
  },  
]
```

User Data (Example API Response)

```
{  
  "id":1,  
  "name": "Sana Moin",  
  "email": sana.moin@mail.com,  
  "orders":[  
    {  
      "ordered:"345",  
      "date":"20205-01-22",  
      "total": 1500  
    }  
  ]  
}
```


Ecommerce Sample Project



End-Points API

| EndPoint | Method | What it doesn't |
|------------------------------|--------|--|
| /api/rentals/create | POST | Creates a new rental Listing |
| /api/rentals | GET | Fetches all available Listing |
| /api/rentals/[rentalId] | GET | Fetches Detail of a specific rental |
| /api/rentals/[rentalid] | DELETE | Deletes a Specific rental listings |
| /api/rentals/book | POST | Books a rental for a user |
| /api/rentals/user-bookings | GET | Fetches all Bookings made by user |
| /api/payments/create | POST | Initiates payment for rentals |
| /api/payments/verify | GET | Verifies the status of a payment |
| /api/send/confirmation-email | POST | Sends a confirmation email for booking |
| /api/reviews/[rentalId] | GET | Fetches reviews for a specific rental |

