**1. Yêu cầu hệ thống (System Requirements)**

* **Phân tích yêu cầu người dùng**: Tìm hiểu các tính năng mà người dùng cần, như:
  + Đăng ký/đăng nhập tài khoản.
  + Quản lý danh mục sản phẩm nội thất (bàn, ghế, tủ, giường, v.v.).
  + Chức năng giỏ hàng, thanh toán.
  + Xem chi tiết sản phẩm, bình luận và đánh giá.
  + Hỗ trợ lọc và tìm kiếm sản phẩm.
  + Quản lý đơn hàng.
* **Yêu cầu hệ thống**:
  + Tích hợp với các phương thức thanh toán (PayPal, thẻ tín dụng, v.v.).
  + Giao diện thân thiện trên các thiết bị (responsive).
  + Hệ thống phân quyền cho Admin, Khách hàng, Nhân viên.

**2. Thiết kế hệ thống (System Design)**

* **Thiết kế cơ sở dữ liệu (Database Design)**: Dùng MongoDB hoặc MySQL để lưu trữ dữ liệu.
  + **Bảng Users**: Lưu thông tin người dùng (userId, username, password, role, email, phone, address).
  + **Bảng Products**: Lưu trữ sản phẩm (productId, name, description, price, category, stock, imageUrl).
  + **Bảng Categories**: Lưu các danh mục sản phẩm (categoryId, name, description).
  + **Bảng Orders**: Quản lý đơn hàng (orderId, userId, orderDate, totalPrice, status).
  + **Bảng OrderDetails**: Chi tiết đơn hàng (orderDetailId, orderId, productId, quantity, price).
  + **Bảng Reviews**: Quản lý bình luận và đánh giá sản phẩm (reviewId, productId, userId, rating, comment, date).
* **Thiết kế frontend**:
  + Giao diện quản lý cho Admin: Quản lý sản phẩm, đơn hàng, người dùng.
  + Giao diện người dùng: Hiển thị sản phẩm, lọc, tìm kiếm, giỏ hàng, thanh toán.
* **Thiết kế backend**:
  + Sử dụng Node.js kết hợp với Express.js để xây dựng API quản lý sản phẩm, đơn hàng, người dùng.
  + Authentication/Authorization: JWT cho xác thực người dùng và phân quyền.
* **Thiết kế UI/UX**:
  + Tạo wireframe và prototype cho các trang chính: trang chủ, danh sách sản phẩm, chi tiết sản phẩm, thanh toán, giỏ hàng, quản lý đơn hàng.

**3. Phát triển hệ thống (Development)**

* **Frontend (Angular)**:
  + Phát triển các component chính: Header, Footer, ProductList, ProductDetails, Cart, Checkout.
  + Xây dựng service để gọi API (HTTPClient) lấy dữ liệu từ backend.
  + Routing giữa các trang: Home, Product Details, Cart, Checkout.
  + Reactive Forms cho đăng ký, đăng nhập, và checkout.
* **Backend (Node.js + Express.js)**:
  + Tạo các API cần thiết cho CRUD (Create, Read, Update, Delete) sản phẩm, đơn hàng, người dùng.
  + Xử lý các yêu cầu liên quan đến giỏ hàng, thanh toán.
  + Quản lý phân quyền và xác thực với JWT.
* **Database**:
  + Cài đặt MongoDB hoặc MySQL.
  + Viết các query để lưu trữ, cập nhật, và truy vấn dữ liệu.

**4. Kiểm thử (Testing)**

* **Kiểm thử đơn vị (Unit Testing)**: Sử dụng Jasmine hoặc Jest để kiểm tra các module riêng lẻ (frontend/backend).
* **Kiểm thử tích hợp (Integration Testing)**: Đảm bảo các phần của hệ thống hoạt động cùng nhau một cách mượt mà.
* **Kiểm thử hệ thống (System Testing)**: Kiểm tra toàn bộ hệ thống, từ giao diện người dùng đến backend và cơ sở dữ liệu.
* **Kiểm thử bảo mật (Security Testing)**: Đảm bảo dữ liệu người dùng được bảo vệ, tránh SQL Injection, CSRF, và XSS.

**5. Triển khai (Deployment)**

* **Triển khai Backend**: Dùng Heroku, AWS, hoặc một nền tảng cloud khác để triển khai Node.js server.
* **Triển khai Frontend**: Dùng Netlify hoặc Vercel để triển khai ứng dụng Angular.
* **Cơ sở dữ liệu**: Dùng MongoDB Atlas (cho MongoDB) hoặc Amazon RDS (cho MySQL) để lưu trữ cơ sở dữ liệu trên cloud.
* **CD/CI (Continuous Deployment/Continuous Integration)**: Tích hợp Jenkins hoặc GitHub Actions để tự động triển khai và kiểm thử.

**6. Bảo trì (Maintenance)**

* **Giám sát hệ thống**: Theo dõi lỗi và hiệu suất của website qua các công cụ như New Relic, Google Analytics.
* **Cập nhật tính năng**: Dựa trên phản hồi từ người dùng, tiếp tục phát triển và cải thiện hệ thống.
* **Bảo trì bảo mật**: Thường xuyên cập nhật các phiên bản mới của framework và các gói phần mềm để bảo đảm an toàn.

Kế hoạch này theo mô hình Waterfall, nên các giai đoạn cần hoàn thành tuần tự trước khi chuyển sang giai đoạn tiếp theo.