

Zeenet E-commerce Web Application Documentation

Overview

The Zeenet e-commerce web application is a MERN-stack web app consisting of three separate apps: Zeenet Frontstore, Zeenet Adminapp, and Zeenet Backend. Each app is deployed independently on Render, providing a scalable and maintainable e-commerce platform.

Components

Zeenet Frontstore

- **Description:** The Zeenet Frontstore is the client-facing application, providing customers with a seamless and intuitive interface to browse and purchase products.
- **Features:**
 - Product browsing and searching
 - Product details and reviews
 - Shopping cart and checkout functionality
 - Payment gateway integration with M-pesa.
 - User authentication and authorization
- **Technologies:**
 - **React:** For building the user interface.
 - **Redux:** For state management.
 - **Bootstrap:** For styling.
 - **Axios:** For making HTTP requests.
 - **Ant design tables:** For tables.
 - **React-router-dom:** For app routing.
 - **Formik and yup:** For input validation.
- **Deployment:** <https://zeenet-frontstore.onrender.com>

Zeenet Adminapp

- **Description:** The Zeenet Adminapp is a dashboard for administrators to manage the e-commerce platform, including product management, order management, and customer management.
- **Features:**
 - User authentication and authorization
 - Product creation and management
 - Order management and fulfilment

- Reporting and analytics
- Inventory management
- **Technologies:**
 - **React:** For building the user interface.
 - **Redux:** For state management.
 - **Bootstrap:** For styling.
 - **Axios:** For making HTTP requests.
 - **Ant design tables:** For tables
 - **React-router-dom:** For app routing.
 - **Formik and yup:** For input validation.
- **Deployment:** <https://zeenet-adminapp.onrender.com>

Zeenet Backend

- **Description:** The Zeenet Backend is the server-side application responsible for handling API requests, data storage, and business logic.
- **Features:**
 - RESTful API for data interaction
 - Data storage and retrieval using MongoDB
 - Authentication and authorization using JSON Web Tokens (JWT)
 - Payment gateway integration with M-pesa.
 - Order processing and fulfilment
- **Technologies:**
 - **Node.js:** For server-side JavaScript execution.
 - **Express.js:** For building the REST API.
 - **MongoDB:** For the database.
 - **Mongoose:** For MongoDB object modeling.
- **Deployment:** <https://zeenet-backend.onrender.com>

API Documentation

- **API Endpoints:** <https://zeenet-backend.onrender.com>

Database Schema

The Zeenet Backend uses MongoDB as the database management system. The database schema is designed to store and manage e-commerce data, including:

- **Products:** product information, pricing, and inventory
- **Orders:** order information, customer details, and payment status
- **User:** customer information, order history, and loyalty program data
- **Payments:** payment information, transaction history, and payment status
- **Product Categories**
- **Blogs**
- **Blog Categories**
- **Brands**
- **Cart**
- **Coupon**
- **Enquiry**

Security

The Zeenet e-commerce web application follows best practices for security, including:

- **Authentication and Authorization:** Using JSON Web Tokens (JWT) for authentication and authorization.
- **Data Encryption:** Using HTTPS for data encryption.
- **Input Validation:** Validating user input using formik and yup to prevent SQL injection and cross-site scripting (XSS).
- **Payment Gateway Integration:** Using secure payment gateways with M-pesa to process payments.

Development

Prerequisites

- Node.js
- MongoDB
- Render account

To contribute to the Zeenet e-commerce web application, follow these steps:

1. Clone the repository for each app using Git.
 1. git clone <https://github.com/edwinonyango9286/Zeenet-Adminapp.git>
 2. git clone <https://github.com/edwinonyango9286/Zeenet-Backend.git>
 3. git clone <https://github.com/edwinonyango9286/Zeenet-Frontstore.git>
2. Set up environment variables by creating a .env file and add the following configurations.

Zeenet App Configurations.

Zeenet port number.

PORT = 8000

Zeenet connection to the database.

MONGODB_URL=""

Zeenet jwt secret

JWT_SECRET=""

Zeent email configuration.

MAIL_ID=""

MP=""

Zeenet cors url for backend access

ORIGIN=["http://localhost:3000","http://localhost:3001","http://localhost:8000"]

Zeenet cloudinary configuration.

CLOUD_NAME=""

API_KEY=""

API_SECRET=""

Zeenet safaricom daraja api intergration

TILL=

STORE=

SAFARICOM_API_KEY=

SAFARICOM_API_SECRET=

SAFARICOM_SHORTCODE=

SAFARICOM_PASSKEY=

3. Install dependencies using npm install.
4. Start the development server for the backed using npm run dev.

5. Make changes to the codebase and commit using Git.
6. Push changes to the remote repository.

Testing

The Zeenet e-commerce web application uses Jest and Enzyme for unit testing and integration testing.

- **Zeenet Frontstore:**

npm run test

- **Zeenet Adminapp:**

npm run test

- **Zeenet Backend:**

npm run test

Usage

Zeenet Frontstore

1. Access the frontstore application via its deployed URL <https://zeenet-frontstore.onrender.com>
2. Register or log in to your account.
3. Browse products, add them to the cart, and place orders.

Zeenet AdminApp

1. Access the admin application via its deployed URL: <https://zeenet-adminapp.onrender.com/admin>
2. Log in with admin credentials.
3. Manage products, orders, and users.

Zeenet Backend

1. The backend provides API endpoints for the frontend applications.
2. Ensure the backend server is running and accessible via its deployed URL.

Code Structure

The codebase for each app is organized using a modular and scalable architecture.

- **Zeenet Frontstore:**

Markdown

cd frontend

1. public

2. src
 - a. App
 - b. Components
 - c. Features
 - d. images
 - e. Pages
 - f. routing
 - g. utils
 - h. App.css
 - i. App.js
 - j. index.js
 - k. .gitignore
 - l. package-lock.json
 - m. package.json

3. LICENSE

4. README.md

- **Zeenet Adminapp:**

Markdown

cd adminapp

1. public
2. src
 - a. App
 - b. Components
 - c. features
 - d. pages
 - e. images
 - f. routing
 - g. utils
 - h. app.css
 - i. app.js
 - j. index.css
 - k. index.js
 - l. gitignore
 - m. package-lock.json
 - n. package.json

3. LICENSE

4. README.md

- **Zeenet Backend:**

Markdown

cd backend

1. config
2. models
3. middlewares

4. controllers
5. public
6. routes
7. services
8. utils
9. index.js
10. LICENSE
11. README.md

Conclusion

The Zeenet e-commerce web application is a scalable and maintainable MERN-stack web application, consisting of three separate apps: Zeenet Frontstore, Zeenet Adminapp, and Zeenet Backend. Each app is deployed independently on Render, providing a seamless and intuitive user experience. This documentation provides an overview of the application's components, features, and technologies, as well as guidelines for development, testing, and security.