E-commerce Website

**Project Overview**

**Tech Stack**: React (Frontend) with Vite – Express (Backend) – MongoDB (Database) – Node.js – Tailwind CSS (Styling)

**Description**

The **E-commerce Inventory Management App** is a powerful tool designed to streamline product tracking across an e-commerce platform. This app centralizes inventory data and provides users with intuitive CRUD (Create, Read, Update, Delete) functionalities to maintain accurate records of each item. Users can easily manage product information, including Product ID, Product Name, Category, Stock Quantity, Price. Built with a robust tech stack, the app ensures that inventory data remains current, enhancing the efficiency and reliability of the e-commerce business.

**Table of Contents**

1. [Project Structure](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#project-structure)
2. [Database Design](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#database-design)
3. [API Endpoints](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#api-endpoints)
4. [Frontend Components](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#frontend-components)
5. [Backend Architecture](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#backend-architecture)
6. [Additional Functionalities](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#additional-functionalities)
7. [Setup and Installation](file:///C:\Users\anilb\AppData\Local\Microsoft\Windows\INetCache\IE\EBFVXFEY\Flight_App%5b1%5d.docx#setup-and-installation)

E- commerce app /

── backend/

── controllers/

── product.js

── auth.js

── models/

── product.js

── user.js

── routes/

── productrout.js

── authrout.js

── server.js

──. env

── frontend/

── public/

── src/

── components/

── App.jsx

── index.html

── package. Json

* backend/: Contains the Node.js and Express.js API server
  + controllers/: Handles business logic for CRUD operations
  + models/: Mongoose models for MongoDB schema
  + routes/: API routes for trainer CRUD operations
* frontend/: Contains the React application for UI
  + components/: Reusable UI components
  + src/: page components

**Database Design**

**Collections:** E-commerce

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| \_id | Object | Unique identifier for each flight |
| name | String | Product name |
| description | String | Product description |
| price | String | Product price |
| image | Number | Product image |
| category | Number | Category of product |

**3. API Endpoints**

**Base URL:** http://localhost:3000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Endpoint** | **Description** | **Request Body** | **Response** |
| GET | /api/products/ | Fetch all products | - | Array of product |
| GET | /category/:category | Fetch specific products | - | Products |
| POST | /api/products/ | Add new product | { name, description, price, image, category} | Creates new flight |
| PUT | /api/products/:id | Update product by ID | { name, description, price, image, category} | Updates product details in the list |
| DELETE | /api/products/:id | Delete product by ID | - | product detail will be removed from list |

**Frontend Components**

**Pages**

1. **Homepage** (/)
   * Displays a list of all producys with add, edit and delete buttons.
2. **Add Flight Page** (/Add Flight)

Contains a form for adding a new trainer with fields { name, description, price, image, category}

**Edit Flight Page** (/api/products/:id)

* + Form pre-filled with existing flight data for updating.

