ONLINE HEALTH AND BEAUTY PRODUCT ORDERING SYSTEM

“Golden Aura” System

Software Installation Guide

Ganhewage GDM

E1946325

Faculty of Information Technology

University of Moratuwa

December 2024

Table of Contents

This software installation guide provides step-by-step instructions for installing and setting up this online health and beauty product ordering system on your local machine.

1. Prerequisites

Before installing this online health and beauty product ordering system, make sure the following installed on your system:

- MongoDB: MongoDB Compass (Download - https://www.mongodb.com/try/download/compass)
- Node.js: Version 20.11.1 x or later (Download - https://nodejs.org/en/download)
- VS Code or any code editor (Download - https://code.visualstudio.com/)
- Cloudinary Account: Required for image uploads (Create Account- https://cloudinary.com/home)
- Stripe account: Required for payment (Create Account - https://dashboard.stripe.com/login)
- Git: (Download - https://git-scm.com/)

2. Download the project

You can download the project using Google Drive or GitHub.

Option 1: Google Drive

- Download the project from Google Drive (ZIP file)

https://drive.google.com/file/d/1PrC5wjP3tQ7LY94kSgv0zpXP_xERuCzr/view?usp=sharing

- After downloading, Extract the ZIP file to your preferred location

Option 2: GitHub

- Open a terminal and run the following command

git clone https://github.com/DManjula98/Golden_Aura_System_Full_Code-E1946325.git

- Navigate to the project folder.

Type → cd project-folder

3. Install Dependencies

3.1. Backend

- Open the project in VS
- Open the Terminal
- Type → cd backend (To go to the backend folder)
- Type → npm install

3.2. Consumer Dashboard

- Add the new Terminal
- Type → cd consumer (To go to the consumer folder)
- Type → npm install

3.3. Frontend (seller and admin)

- Add the new Terminal
- Type → cd frontend (To go to the frontend folder)
- Type → npm install

4. Setup Environment Variables

4.1. backend (.env)

- Open the backend folder
- Locate and open the sample **.env** file
- Update the following:
    - PORT = 5000
    - DB_URL = your-MongoDB-connection-string [Check Topic: 5]
    - SECRET = your-secret-key
    - CLOUD_NAME = your-cloud-name [Check Topic: 7]
    - CLOUD_API_KEY = your-api-key [Check Topic: 7]
    - CLOUD_API_SECRET = your-api-secret [Check Topic: 7]
    - stripe_Key = your-stripe-key [Check Topic: 8]
    - GOOGLE_user = your-email (admin)
    - GOOGLE_password = your-google-password [Check Topic: 9]
    - CLIENT_URL = http://localhost:3000

The required format is already provided by the sample .env file in the backend directory. Provide the CLOUD_NAME, CLOUD_API_KEY, and CLOUD_API_SECRET of your Cloudinary account instead of the currently provided CLOUD_NAME, CLOUD_API_KEY, and CLOUD_API_SECRET. Also provide the Stripe_Key of your Stripe account instead of the currently provided Stripe_Key. Furthermore, add the GOOGLE_user and GOOGLE_password of your Gmail account to this .env file instead of the currently provided GOOGLE_user and GOOGLE_password.

4.2. consumer (./../utils/config.js)

- Open the consumer folder
- Navigate to the src folder
- Open the utils folder inside src
- Locate and open the sample **config.js** file
- Update the following:
    - Stripe_sky = your-publishable-key [Check Topic: 8]

Add your Stripe account's publishable key here instead of the publishable key currently provided. (Stripe_sky)

5. **Create Database using MongoDB Compass**
    - Open MongoDB Compass
    - Connect to your local MongoDB instance.
        - Click Add new connection
        - Click Connect button
    - Click “Create Database”
    - Name the database: (ex-: Golden_Aura)
    - Inside the .env file, update
        - DB_URL = mongodb://localhost:27017/Golden_Aura
    - Create collection inside the Golden_Aura database
        - admins
        - authorders
        - banners
        - blogs
        - carts
        - categories
        - customers
        - myshopwallets
        - orders
        - products
        - reviews
        - seller_admin_messages
        - seller_customer_messages
        - seller_customers
        - sellers
        - sellerwallets
        - stripes
        - wishlists
        - withdrowrequests

6. **Import Sample Data in to the MongoDB (Products, Categories, Blogs, Admin, Sellers, and Stripes data)**
    - Open MongoDB Compass and connect to mongodb://localhost:27017
    - Select your database (Golden_Aura)
    - Click on the collection. (Products, Categories, Blogs, Admin, Sellers, and Stripes data)
    - Import provided sample JSON files
        - Click “Import Data”
        - Select the .json file (provided in the “Golden Aura System_Source Code" folder)
            - Golden_Aura.product.json – Import into products collection
            - Golden_Aura.categories.json - Import into categories collection
            - Golden_Aura.bolgs.json – Import into blogs collection
            - Golden_Aura.banners.json – Import into banners collection
            - Golden_Aura.sellers.json – Import into sellers collection
            - Golden_Aura.seller_customers.json – Import into seller_customers collection
            - Golden_Aura.admins.json – Import into admin collection
            - Golden_Aura.stripes.json – Import into stripes collection

Here, the **admin collection of the above database needs to be imported**. But **adding data to other collections is not mandatory**; they can be done by the admin and sellers through their dashboards. But adding that collection will make it easier to use this system. (For example, health and beauty products can be seen through this system only after the seller adds the health and beauty products to this system.)

7. Cloudinary Setup

This project uses Cloudinary for image storage. Follow these steps to configure it.

- Create a Cloudinary Account:
    - Go to Cloudinary
    - Sign up for a free account.
    - After signing up, go to the Dashboard to find your API credentials.
- Navigate to Dashboard and Get the following Credentials:
    - Cloude Name
    - API Key
    - API Secret
- Add them to .env file in backend:
    - CLOUD_NAME = your-cloud-name
    - CLOUD_API_KEY = your-api-key
    - CLOUD_API_SECRET = your-api-secret
- Create Folder for upload images
    - Go to the Media Library
    - Select Folder option
    - Create new folders with the following names:
        - admin_profiles - Upload admin profile images to the ‘admin_profiles’ folder in Cloudinary.
        - blogs - Upload blog images to the ‘blogs’ folder in Cloudinary.
        - profiles - Upload sellers profile images to the ‘profiles’ folder in Cloudinary.
        - banners - Upload banner images to the ‘banners’ folder in Cloudinary.
        - categories - Upload category images to the ‘categories’ folder in Cloudinary.
        - products - Upload product images to the ‘products’ folder in Cloudinary.
        - subcategories - Upload subcategory images to the ‘subcategories’ folder in Cloudinary.

8. **Stripe Payment Integration**
    - Sign up on Stripe
    - Navigate to Developers → API Keys
    - Copy your Publishable Key and Secret Key
    - Add stripe_Key to the backend .env file:
        - stripe_Key = your-stripe-secret-key
    - Add stripe_sky to the consumer config.js file:
        - stripe_sky = you-publishable-key

9. **Generating a Google App Password for Email Services**
    - Go to Google Account Security.
    - Enable 2-Step Verification.
        - Scroll down to “Signing in to Google”
        - Click 2-Step Verification and follow the step process.
    - Generate App Password
        - Search App Password
        - Give App name and click Create button.
    - Copy and Save the App Password
        - Google will generate a 16-character app password
        - Copy this password
    - Add this password to .env file in backend:
        - GOOGLE_user = your-email [admin]
        - GOOGLE_password = your-google-password

10. Run the Project Locally

10.1. Start the Backend Server (API)

- Type → cd backend (If you are already in the backend folder, you don't need to type this, but if you are not in the backend folder, type this.)
- Type → nodemon server.js

The backend will run at htttp://localhost:5000

10.2. Start Consumer Dashboard

- Type → cd consumer (If you are already in the consumer folder, you don't need to type this, but if you are not in the consumer folder, type this.)
- Type → npm start

The consumer will run at http://localhost:3000

10.3. Start Frontend (Admin and Seller Dashboard)

- Type → cd frontend (If you are already in the frontend folder, you don't need to type this, but if you are not in the frontend folder, type this.)
- Type → npm start
- Terminal give this message “Would you like to run the app on another port instead? » (Y/n)”
- Type → y

The Seller Dashboard will run at - http://localhost:3001/login

- Open another web browser
- Give this URL - http://localhost:3001/admin/login

The Admin Dashboard will run at - http://localhost:3001/admin/login {important: don’t access in same web browser}

11. Access the System

Once the servers are running:

- Consumer Dashboard: http://localhost:3000
- Seller Dashboard: http://localhost:3001/login
- Admin Dashboard: http://localhost:3001/admin/login

11.1. Admin Login Credentials:

- Email: dilrukshimanjula14@gmail.com
- Password: 123456

11.2. Sellers Login Credentials:

- Email: amalierandika22@gmail.com
- Password: 123456

- Email: sugathadasa56@gmail.com
- Password: 123456

By following these steps, you will be able to easily install this online health and beauty product ordering system on your local machine. If you encounter any problems, please contact me through the methods given below.

- Contact me from: -
    - Email - e1946325@bit.mrt.ac.lk
    - Mobile – 0769364744