Project Proposal: Press-On Nails E-Commerce Platform

1. What tech stack will you use for your final project?

- **Tech Stack**:
    - **Front-End**: React.js
    - **Back-End**: Node.js with Express
    - **Database**: MongoDB
    - **Deployment**: Heroku (Back-End) and Vercel (Front-End)
    - **Payment Gateway**: Stripe API for handling payments

2. Is the front-end UI or the back-end going to be the focus of your project? Or are you going to make an evenly focused full-stack application?

- This project will be an evenly focused full-stack application. Both the front-end and back-end will be developed with equal emphasis to ensure a seamless user experience and efficient data management.

3. Will this be a website? A mobile app? Something else?

- This will be a responsive website, optimized for both desktop and mobile devices, providing a wide reach and accessibility to users.

4. What goal will your project be designed to achieve?

- The goal of this project is to create a specialized e-commerce platform for selling press-on nails. The site will offer a curated collection of high-quality press-on nails and aim to provide an easy-to-use shopping experience with secure payment processing and reliable order tracking.

5. What kind of users will visit your app? In other words, what is the demographic of your users?

- The primary users of the platform will be fashion-conscious individuals, predominantly women aged 18-45, who are interested in convenient and affordable beauty products. The platform will cater to both casual shoppers and those looking for press-on nails for special occasions.

6. What data do you plan on using? How are you planning on collecting your data?

- **Data Types**:
    - **Product Data**: Information on available press-on nail designs, including details like colors, finishes, sizes, and pricing. This data will be manually entered and managed in MongoDB.

    - **User Data**: Information such as user profiles, order history, and wishlists will be securely stored and managed.
    - **Payment Data**: Payment information will be processed through the Stripe API, ensuring secure transactions without storing sensitive payment details on the platform.
- **API Usage**:
    - **Stripe API**: For payment processing.
    - **Shipping API**: Possibly integrate with EasyPost or Shippo for shipping and tracking services.
- **Approach to Data Collection**:
    - The product data will be curated and uploaded to the database. User data will be collected through the registration and purchase process, and payment data will be handled securely via the Stripe API.

Additional Questions:

- **What does your database schema look like?**
    - **Users Collection**: Fields for user ID, email, password (hashed), order history, and wishlist.
    - **Products Collection**: Fields for product ID, name, description, price, available sizes, colors, stock quantity, and image URLs.
    - **Orders Collection**: Fields for order ID, user ID, product IDs, order status, payment status, and shipping information.
- **What kinds of issues might you run into with your API?**
    - **Payment Gateway Integration**: Ensuring smooth integration with Stripe for secure transactions.
    - **Data Security**: Protecting user data, particularly personal information and payment details, will be crucial. Ensuring compliance with regulations like GDPR is necessary.
- **Is there any sensitive information you need to secure?**
    - **User Passwords**: Passwords will be hashed using bcrypt before storing them in the database.
    - **Payment Information**: Payment data will not be stored on the platform; instead, all transactions will be processed securely via the Stripe API.
- **What functionality will your app include?**
    - **Product Browsing and Filtering**: Users can browse products by category and use filters to narrow down their choices.
    - **User Accounts**: Users can create accounts to save their preferences, view order history, and manage wishlists.
    - **Secure Checkout and Payment**: A streamlined checkout process with secure payment options.
    - **Order Tracking**: Users can track the status of their orders from purchase to delivery.

  - **Nail Care Resources**: A section dedicated to nail care guides and styling tips.
- **What will the user flow look like?**
  - **Home Page**: Users land on the home page and browse featured products or navigate to specific collections.
  - **Product Browsing**: Users explore the product catalog, apply filters, and select items of interest.
  - **Product Page**: Users view detailed product information and add items to their cart.
  - **Checkout**: Users review their cart, enter shipping details, and make a payment.
  - **Order Confirmation**: Users receive confirmation of their order and can track it via their account page.
- **What features make your site more than a CRUD app? What are your stretch goals?**
  - **Beyond CRUD**:
    - Integration with Stripe for secure payment processing.
    - User-generated content such as reviews and ratings.
    - Real-time order tracking and email notifications.
  - **Stretch Goals**:
    - **Discount Codes**: Implement a discount code system for promotions.
    - **Recommendation Engine**: Suggest products to users based on their browsing history and past purchases.