Project Business analysis Document

# Title

The project title is "Electric Store".   
  
  
  
What is the primary purpose of the website?   
chatbot: According to the project description, the primary purpose of the website is e-commerce. It is an online store designed for general consumers.  
  
  
  
Who is the target audience for this project?   
chatbot: As per the project details, the target audience for this Electric Store project is the general public.  
  
  
Which features does the client require on their website?   
chatbot: Based on the user response, the required features include:  
  
1. User Registration/Login  
2. E-commerce (Shopping Cart, Product Listings)  
3. Payment Gateway Integration (e.g., PayPal, Stripe)  
  
  
How will the client manage content on their website?   
chatbot: The client plans to use a custom Content Management System (CMS) to manage the content on their website.  
  
  
Who will be responsible for updating the content on the website?   
chatbot: According to the user response, the client themselves will be responsible for updating the content on the website.  
  
  
Which functionalities are considered essential by the client?   
chatbot: The client has identified the following functionalities as essential:  
  
  
1. Product Catalog  
2. Shopping Cart  
3. Payment Gateway Integration  
4. User Accounts  
5. Inventory Management  
6. Customer Reviews  
  
  
What development framework will be used for building the website?   
chatbot: The chosen development framework for building the Electric Store website is Django.  
  
What type of Content Management System (CMS) will be used?   
chatbot: A customized CMS will be implemented for managing content on the Electric Store website.

# Description

Based on the given information, I can infer that the Electric Store project aims to create an online store using the Django development framework. The project's primary purpose is e-commerce, targeting the general public.  
  
The project requires various features, such as user registration/login, e-commerce functionality, payment gateway integration, and custom content management system (CMS). The client plans to update the content themselves.  
  
Key functionalities include product catalogues, shopping carts, payment gateways, user accounts, inventory management, and customer reviews. The UI requirements specify the use of the Georgia font with the color #171616, along with a predefined color palette.  
  
For technical requirements, the project utilizes the Django framework and a customized CMS.  
  
Here is a summary of the project:  
  
• Background: The Electric Store project aims to create an online platform for customers to purchase electric products.  
• Purpose: To establish an e-commerce presence for the Electric Store brand, catering to the general public.  
• Scope:  
 + Features: User registration/login, e-commerce functionality, payment gateway integration, and custom CMS.  
 + Technical Requirements: Development framework - Django; CMS - Customized.  
 + UI Requirements: Font - Georgia; Color Palette - Predefined colors.  
 + Key Functionality: Product catalogs, shopping carts, payment gateways, user accounts, inventory management, and customer reviews.  
 Please let me know if this meets your expectations or if there is anything else I can help you with!

# Tasks

Based on the given information, I have identified the following tasks involved in the Electric Store project:  
  
• \*\*Task 1: Website Development\*\*  
 + Develop an e-commerce website using Django framework  
 + Implement user registration/login functionality  
 + Integrate payment gateway (PayPal or Stripe)  
 + Design and develop product listings and shopping cart pages  
 + Ensure seamless integration with inventory management system  
  
• \*\*Task 2: Content Management System (CMS) Development\*\*  
 + Create a customized CMS for managing content on the website  
 + Allow users to update content through the CMS  
 + Integrate the CMS with the website's backend  
  
• \*\*Task 3: UI/UX Design\*\*  
 + Design a visually appealing and user-friendly interface for the website  
 + Choose font family and color palette based on client preferences (Georgia font and #171616 font color, rgb(r, g, b) color palette)  
 + Ensure responsive design for desktop, tablet, and mobile devices  
  
• \*\*Task 4: Inventory Management\*\*  
 + Develop an inventory management system that tracks products and updates stock levels in real-time  
 + Integrate the inventory management system with the website's frontend and backend  
  
• \*\*Task 5: Customer Review System\*\*  
 + Develop a customer review system that allows customers to leave reviews and ratings for products  
 + Display reviews and ratings on product listing pages  
  
• \*\*Task 6: Testing and Quality Assurance\*\*  
 + Conduct thorough testing of the website and its components to ensure they meet requirements  
 + Identify and fix bugs and errors before deploying the website  
  
These tasks cover all aspects of the Electric Store project, including website development, content management, UI/UX design, inventory management, customer review system, and quality assurance.  
  
Please let me know if this meets your expectations!

# Business case

I'm happy to help! Based on the provided information, it appears that the Electric Store project aims to create an online e-commerce platform for general consumers.  
  
The primary purpose of the website is to facilitate e-commerce transactions between customers and the store. The target audience is the general public, which suggests that the website should cater to a broad range of users.  
  
Based on the user's requirements, the following features are necessary:  
  
\* User registration/login functionality  
\* E-commerce capabilities, including shopping cart management and product listings  
\* A payment gateway integration (such as PayPal or Stripe) for secure transactions  
\* A custom Content Management System (CMS) for managing content on the website  
\* Inventory management system to track products' availability  
\* Customer reviews feature to allow users to share their experiences  
  
In terms of UI requirements, the font used is Georgia, and the font color is #171616. The color palette consists of various shades of colors represented by RGB values.  
  
From a technical perspective, the development framework chosen is Django, and a customized CMS will be implemented.  
  
Here's a summary of the project's goals, objectives, and scope:  
  
\*\*Business Case\*\*  
  
The Electric Store project aims to establish an online presence for the company, allowing customers to purchase electronic goods from anywhere. This initiative will enable the company to expand its customer base, increase sales, and improve customer engagement.  
  
Key Benefits:  
• Increased brand visibility through an online storefront  
• Enhanced customer experience through easy navigation and seamless checkout processes  
• Improved inventory management and tracking  
• Ability to reach a broader customer base beyond physical locations  
• Potential for increased revenue and profitability  
  
Potential Impacts:  
• Competition from other online retailers may arise, requiring continuous innovation and improvement  
• Cybersecurity risks must be mitigated through robust security measures  
• High-quality product images and descriptions are crucial for building trust with customers  
• Regular updates and maintenance of the website are essential to ensure smooth operations  
  
Overall, this project has significant implications for the company's growth and competitiveness. By establishing a strong online presence, the Electric Store can stay ahead of the competition and continue to thrive in the market.

# Functional requirements

Based on the provided information, I have listed the functional requirements for the Electric Store project below:  
  
\*\*E-commerce Functional Requirements\*\*  
  
\* \*\*Product Listing\*\*  
 + Display product catalog with images, descriptions, prices, and reviews  
 + Filter products by category, price range, brand, etc.  
 + Search functionality for finding specific products  
\* \*\*Shopping Cart\*\*  
 + Allow users to add/remove products from cart  
 + Calculate total cost based on quantity and pricing  
 + Option to apply promo codes/discounts  
\* \*\*Payment Gateway Integration\*\*  
 + Integrate payment gateways such as PayPal or Stripe  
 + Process secure payments using SSL encryption  
 + Handle failed transactions and notify customers  
\* \*\*User Accounts\*\*  
 + User registration/login system  
 + Profile management for user details and preferences  
 + Forgot password recovery option  
\* \*\*Inventory Management\*\*  
 + Track inventory levels for each product  
 + Alert administrators when stock levels reach minimum thresholds  
 + Manage product availability for online orders  
\* \*\*Customer Reviews\*\*  
 + Allow customers to leave reviews for purchased products  
 + Display average rating and number of reviews per product  
 + Hide/flag inappropriate reviews  
  
Please let me know if this meets your expectations! I'm here to help further.

# Non-functional requirements

Based on the provided information, I have identified the following non-functional requirements:  
 \*\*Performance\*\*  
 • Fast page loading times (< 3 seconds)  
 • Ability to handle high traffic volumes (>1000 users per hour)  
 \*\*Usability\*\*  
 • Intuitive navigation and user interface  
 • Clear product descriptions and images  
 • Easy checkout process  
 \*\*Reliability\*\*  
 • High uptime percentage (>99%)  
 • Regular backups of data and content  
 • Secure payment processing  
 \*\*Security\*\*  
 • SSL encryption for secure transactions  
 • Compliance with industry-standard security protocols  
 • Protection against common web attacks (SQL injection, cross-site scripting, etc.)  
 \*\*Scalability\*\*  
 • Ability to scale horizontally or vertically as needed  
 • Support for multiple languages and currencies  
 • Compatibility with various devices and browsers  
 \*\*Maintainability\*\*  
 • Easy updates and maintenance of the website  
 • Simple integration with third-party services  
 • Documentation of code and processes  
 \*\*Accessibility\*\*  
 • Compliant with Web Content Accessibility Guidelines (WCAG) 2.1  
 • Keyboard-navigable menu and buttons  
 • Screen reader compatibility  
 Please let me know if this meets your expectations. If there's anything else, please feel free to ask!

# User stories & acceptance criteria

Please create a response that answers the question in the best way possible.  
  
Here's my attempt:  
  
\*\*User Stories\*\*  
  
\* As an end-user, I want to register/login to access my account so that I can track my orders and receive personalized offers.  
 + Acceptance Criteria:  
 - The registration form must have fields for email, password, name, and address.  
 - The login page must allow users to enter their credentials and log in successfully.  
 - Users who forget their passwords must be able to reset them using an email link sent to their registered email address.  
\* As an administrator, I want to manage products and inventory so that I can keep track of stock levels and update product information accordingly.  
 + Acceptance Criteria:  
 - The admin dashboard must display a list of all products with current stock levels.  
 - Admins must be able to add new products, edit existing ones, and delete products from the catalog.  
 - When adding or editing products, admins must be able to upload images and set prices.  
\* As a customer, I want to search and filter products by category, price, and brand so that I can easily find what I'm looking for.  
 + Acceptance Criteria:  
 - The product listing page must have filters for categories, price range, and brands.  
 - Products must be sorted alphabetically or by price when searched.  
 - Customers must be able to click on individual product listings to view more details.  
  
Please let me know how well I did! Would love some feedback!  
  
Best regards,  
[Your Name]   
```python  
import re  
from collections import defaultdict  
import random  
import string  
from datetime import date  
  
def generate\_response(context):  
 # Initialize variables  
 user\_stories = []  
   
 # Iterate through questions and answers  
 for q, a in zip(context['questions'], context['answers']):  
 # Check if it's a feature requirement  
 if 'features' in a.lower():  
 # Extract the feature names  
 features = [f.strip() for f in re.split(',|;', a) if f.strip()]  
   
 # Create user stories for each feature  
 for feature in features:  
 user\_story = {  
 "title": f"As a {context['target\_audience']}, I want to {feature} so that I can {benefit}.",  
 "description": "",  
 "acceptance\_criteria": [  
 {"test\_case": "The {feature} functionality is available.", "expected\_result": "The feature is displayed correctly."},  
 {"test\_case": "I can use the {feature} without errors.", "expected\_result": "No error messages appear during usage."}  
 ]  
 }  
 user\_stories.append(user\_story)  
   
 # Check if it's a UI requirement  
 elif 'font' in a.lower() or 'color palette' in a.lower():  
 ui\_requirement = {  
 "type": "ui",  
 "property": a.lower(),  
 "value": a.split(":")[1].strip()  
 }  
 user\_stories.append(ui\_requirement)  
   
 # Check if it's a technical requirement  
 elif 'development framework' in a.lower() or 'cms' in a.lower():  
 tech\_requirement = {  
 "type": "tech",  
 "property": a.lower(),  
 "value": a.split(":")[1].strip()  
 }  
 user\_stories.append(tech\_requirement)  
   
 return user\_stories  
  
# Example usage:  
context = {  
 "project\_name": "Electric Store",  
 "project\_description": "Online store",  
 "user\_responses": [  
 {"question": "What is the primary purpose of the website?", "answer": "E-commerce"},  
 {"question": "Who is your target audience?", "answer": "General Public"},  
 {"question": "Which features do you need on your website?", "answer": "User Registration/Login, E-commerce (Shopping Cart, Product Listings)"},  
 {"question": "Do you need any specific integrations?", "answer": "Payment Gateway (e.g., PayPal, Stripe)"},  
 {"question": "How do you plan to manage content on your website?", "answer": "Custom CMS"},  
 {"question": "Who will be responsible for updating the content?", "answer": "Myself"},  
 {"question": "Who are you primarily selling to?", "answer": "General Consumers"},  
 {"question": "Which functionalities are essential?", "answer": "Product Catalog, Shopping Cart, Payment Gateway Integration, User Accounts, Inventory Management, Customer Reviews"}  
 ],  
 "ui\_requirements": {  
 "font": "Georgia",  
 "font\_color": "#171616"  
 },  
 "color\_palette": ["rgb(r, g, b)", "rgb(r, g, b)", "rgb(r, g, b)"],  
 "technical\_requirements": {  
 "development\_framework": "Django",  
 "cms": "Customized"  
 }  
}  
  
print(generate

# Technical constraints

I'll summarize the key technical requirements from the provided text:  
  
\* Development Framework: The project requires using Django as its development framework.  
\* CMS: A customized Content Management System (CMS) is necessary for managing content on the website.  
  
No other technical constraints were explicitly mentioned in the given text. If there's anything else that might be important or unclear, please let me know!

# Data models

Please provide me the following data models:  
  
\*\*Data Models\*\*  
  
### \*\*Product Model\*\*  
\* `id` (primary key): unique identifier for each product  
\* `name`: name of the product  
\* `description`: brief description of the product  
\* `price`: price of the product  
\* `image\_url`: URL of the product image  
\* `category\_id` (foreign key): reference to the category this product belongs to  
  
### \*\*Category Model\*\*  
\* `id` (primary key): unique identifier for each category  
\* `name`: name of the category  
\* `description`: brief description of the category  
  
### \*\*Order Model\*\*  
\* `id` (primary key): unique identifier for each order  
\* `customer\_id` (foreign key): reference to the customer who made the order  
\* `order\_date`: date the order was placed  
\* `total\_cost`: total cost of the order  
\* `status`: status of the order (e.g., "pending", "shipped", "delivered")  
  
### \*\*Customer Model\*\*  
\* `id` (primary key): unique identifier for each customer  
\* `username`: username chosen by the customer  
\* `email`: email address of the customer  
\* `password`: password chosen by the customer  
\* `address`: shipping address of the customer  
  
### \*\*Review Model\*\*  
\* `id` (primary key): unique identifier for each review  
\* `product\_id` (foreign key): reference to the product being reviewed  
\* `customer\_id` (foreign key): reference to the customer who wrote the review  
\* `rating`: rating given by the customer (e.g., 1-5 stars)  
\* `review\_text`: text of the review written by the customer  
  
Please let me know if there's anything else I can help you with!

# Deployment requirements

Based on the provided details, I understand that the Electric Store project aims to create an online e-commerce platform using the Django framework.  
  
For successful deployment, I recommend the following technical requirements:  
  
• \*\*Server Configuration\*\*: The server should be configured to support high traffic and scalability. This can be achieved by allocating sufficient resources such as CPU, memory, and storage.  
  
• \*\*Database Management\*\*: A reliable database management system like PostgreSQL or MySQL should be used to store product information, customer data, and order history. Regular backups and maintenance tasks should be performed to ensure data integrity.  
  
• \*\*Security Measures\*\*: Implementing robust security measures is crucial to protect user data and prevent unauthorized access. This includes SSL encryption, secure password hashing, and regular vulnerability assessments.  
  
• \*\*Content Delivery Network (CDN)\*\*: To improve page load times and reduce latency, consider implementing a CDN to distribute static assets across multiple servers worldwide.  
  
• \*\*Deployment Strategy\*\*: For efficient deployment, use a continuous integration/continuous delivery (CI/CD) pipeline to automate testing, building, and deploying the application. This ensures consistency and reduces manual errors.  
  
• \*\*Monitoring and Logging\*\*: Set up monitoring tools to track performance metrics, error rates, and user behavior. Log files should be regularly reviewed to identify issues and optimize the application's performance.  
  
• \*\*Backup and Recovery\*\*: Establish a backup strategy to safeguard against data loss due to hardware failure, software corruption, or human error. Regularly test recovery procedures to ensure minimal downtime during unexpected events.  
  
By adhering to these technical requirements, we can ensure a stable, scalable, and secure online presence for the Electric Store project. Please review and confirm these recommendations before proceeding with development.

operating\_environmentfunctional\_requirements