# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Customer Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Customers can register for an account by providing valid personal information, including email and password. The system must validate the email format, password strength, and ensure the email is not already registered.   
\*\*Input:\*\*   
- Name (string)   
- Email (string)   
- Password (string)   
- Optional: Address (string)   
\*\*Output:\*\*   
- Confirmation email sent to the customer's email address   
- Redirect to the login page   
- New customer account created in the system   
  
---  
  
## 1.2 Customer Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Customers can log into their account by entering their email and password. The system must validate the credentials, check if the account is active and verified, and log the login attempt.   
\*\*Input:\*\*   
- Email (string)   
- Password (string)   
\*\*Output:\*\*   
- Redirect to the homepage or account dashboard   
- Login record added to the system log   
- Session created for the logged-in customer   
  
---  
  
## 1.3 Customer Logout Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Customers can securely log out of their account. The system must terminate the session and redirect the customer to the homepage or login page.   
\*\*Input:\*\*   
- Session ID (string)   
- User ID (string)   
\*\*Output:\*\*   
- Session terminated   
- Redirect to the homepage or login page   
- Logout record added to the system log   
  
---  
  
## 1.4 Admin Login Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Administrators can log into the admin interface by entering their username and password. The system must validate the credentials, check if the account is active, and log the login attempt.   
\*\*Input:\*\*   
- Username (string)   
- Password (string)   
\*\*Output:\*\*   
- Redirect to the admin dashboard   
- Login record added to the system log   
- Session created for the logged-in admin   
  
---  
  
## 1.5 Admin Logout Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Administrators can securely log out of their account. The system must terminate the admin session and redirect them to the admin login page or homepage.   
\*\*Input:\*\*   
- Session ID (string)   
- Admin ID (string)   
\*\*Output:\*\*   
- Session terminated   
- Redirect to the admin login page or homepage   
- Logout record added to the system log   
  
---  
  
## 1.6 View Product Details Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Customers can view detailed information about a product, including name, description, price, specifications, and images.   
\*\*Input:\*\*   
- Product ID (string)   
- User ID (string)   
\*\*Output:\*\*   
- Displayed product details (name, description, price, specifications, images)   
- Option to add the product to the cart   
  
---  
  
## 1.7 Add to Cart Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Customers can add products to their shopping cart. The system must check product availability and inventory, and update the cart with the selected product and quantity.   
\*\*Input:\*\*   
- Product ID (string)   
- Quantity (integer)   
- Customer ID (string)   
\*\*Output:\*\*   
- Updated shopping cart with new item and quantity   
- Updated cart summary (total cost, item count)   
- Confirmation message that the product was added   
  
---  
  
## 1.8 Modify Cart Item Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Customers can modify the quantity of an existing item in their cart or remove it. The system must validate the updated quantity against inventory and update the cart summary accordingly.   
\*\*Input:\*\*   
- Cart Item ID (string)   
- New Quantity (integer) or removal request   
- Customer ID (string)   
\*\*Output:\*\*   
- Updated cart item with new quantity or removed item   
- Updated cart summary (total cost, item count)   
- Confirmation message of the modification   
  
---  
  
## 1.9 Remove Cart Item Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Customers can remove a specific item from their shopping cart. The system must verify the existence of the cart item and update the cart summary after removal.   
\*\*Input:\*\*   
- Cart Item ID (string)   
- Customer ID (string)   
\*\*Output:\*\*   
- Cart item removed from the database   
- Updated cart summary (total cost, item count)   
- Confirmation message that the item was removed   
  
---  
  
## 1.10 Checkout Order Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Customers can proceed to checkout with items in their cart. The system must validate the cart, process the selected payment method, update inventory, and send an order confirmation email.   
\*\*Input:\*\*   
- Customer ID (string)   
- Cart ID (string)   
- Payment method (string)   
- Shipping address (string)   
\*\*Output:\*\*   
- Created order record in the database   
- Updated inventory levels   
- Order confirmation email sent to the customer   
- Cart cleared after successful checkout   
  
---  
  
## 1.11 Confirm Order Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* After the payment is processed, the system must confirm the order, update inventory, and send an email confirmation to the customer.   
\*\*Input:\*\*   
- Order ID (string)   
- Payment confirmation status (string)   
- Inventory system access   
\*\*Output:\*\*   
- Finalized order in the database   
- Inventory updated with deducted stock   
- Order confirmation email sent   
- Cart cleared   
  
---  
  
## 1.12 Send Order Confirmation Email Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* The system must send an order confirmation email to the customer after the order is successfully placed. The email must contain order details such as items, total cost, and shipping information.   
\*\*Input:\*\*   
- Order ID (string)   
- Customer email (string)   
- Order details (items, total cost, shipping address)   
\*\*Output:\*\*   
- Order confirmation email sent to the customer   
- Email delivery status logged in the system   
- Updated order status indicating confirmation email was sent   
  
---  
  
## 1.13 Create Product Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Administrators can create a new product by entering its details and uploading documentation or images. The system must validate the input data and save the product in the database.   
\*\*Input:\*\*   
- Product name (string)   
- Description (string)   
- Price (float)   
- Category ID (string)   
- Specifications (string)   
- Documentation/image files   
\*\*Output:\*\*   
- New product record in the database   
- Product displayed in the catalog   
- Confirmation message shown to the administrator   
  
---  
  
## 1.14 Update Product Information Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Administrators can update the information of an existing product, including name, description, price, and category. The system must validate the new data and update the product in the database.   
\*\*Input:\*\*   
- Product ID (string)   
- Updated product name (string)   
- Updated description (string)   
- Updated price (float)   
- Updated category ID (string)   
- Updated documentation/image files (optional)   
\*\*Output:\*\*   
- Updated product record in the database   
- Updated product displayed in the catalog   
- Confirmation message shown to the administrator   
  
---  
  
## 1.15 Manage Inventory Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Administrators can update the inventory levels for products. The system must validate the new inventory quantity and update the database accordingly.   
\*\*Input:\*\*   
- Product ID (string)   
- New inventory quantity (integer)   
\*\*Output:\*\*   
- Updated inventory record in the database   
- Refreshed product catalog reflecting new inventory status   
- Confirmation message shown to the administrator   
  
---  
  
## 1.16 Categorize Products Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Administrators can assign or reassign a product to a specific category. The system must validate the selected category and update the product's category association in the database.   
\*\*Input:\*\*   
- Product ID (string)   
- Category ID (string)   
\*\*Output:\*\*   
- Updated product category in the database   
- Updated product catalog reflecting new categorization   
- Confirmation message shown to the administrator   
  
---  
  
## 1.17 Manage Plugin Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Administrators can install, update, or remove plugins from the system. The system must validate the plugin file and manage its status in the database.   
\*\*Input:\*\*   
- Plugin ID (string)   
- Plugin file or repository link (string)   
- Action (install, update, remove)   
\*\*Output:\*\*   
- Plugin status updated in the database   
- Plugin displayed as active/inactive in the plugin management interface   
- Confirmation message shown to the administrator   
  
---  
  
## 1.18 Access Plugin Documentation Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Administrators can view the documentation for a plugin. The system must retrieve and display the documentation in a structured and user-friendly format.   
\*\*Input:\*\*   
- Plugin ID (string)   
- Admin ID (string)   
\*\*Output:\*\*   
- Plugin documentation displayed (installation instructions, usage examples, etc.)   
- Option to search or navigate within the documentation   
- Error message if documentation is missing or corrupted   
  
---  
  
## 1.19 View Purchase History Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Customers can view their purchase history, which includes all completed orders. The system must retrieve and display order details such as date, items, and total cost.   
\*\*Input:\*\*   
- Customer ID (string)   
\*\*Output:\*\*   
- List of past orders with key details (order date, total cost, status)   
- Detailed view of selected order (items, quantities, shipping address, payment method)   
- Error message if no orders are found   
  
---  
  
## 1.20 Calculate Total Cost Function   
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* The system must calculate the total cost of the cart based on the quantity and price of each item.   
\*\*Input:\*\*   
- Cart Item list (CartItem objects with Product ID and Quantity)   
- Product Price (from Product entity)   
\*\*Output:\*\*   
- Updated TotalCost value for the cart   
- Cart summary updated with the new total cost   
  
---  
  
## 1.21 View Cart Summary Function   
\*\*Function ID:\*\* FR-21   
\*\*Description:\*\* The system must display a summary of the customer's shopping cart, including the number of items and the total cost.   
\*\*Input:\*\*   
- Cart ID (string)   
- Cart Item list (CartItem objects)   
- TotalCost (float)   
\*\*Output:\*\*   
- Cart summary displayed with item count and total cost   
- Error message if cart is empty   
  
---  
  
## 1.22 Manage Quantity Function   
\*\*Function ID:\*\* FR-22   
\*\*Description:\*\* The system must manage the quantity of each item in the cart, ensuring it is valid and does not exceed available inventory.   
\*\*Input:\*\*   
- Cart Item ID (string)   
- New quantity (integer)   
\*\*Output:\*\*   
- Updated quantity in the cart item   
- Updated TotalCost in the cart   
- Error message if the quantity is invalid or exceeds available stock   
  
---  
  
## 1.23 Manage Purchase History Function   
\*\*Function ID:\*\* FR-23   
\*\*Description:\*\* The system must maintain and manage the purchase history of each customer, recording each completed order and its details.   
\*\*Input:\*\*   
- Customer ID (string)   
- Order ID (string)   
- Product ID (string)   
- Purchase date (datetime)   
- Quantity (integer)   
\*\*Output:\*\*   
- Updated PurchaseHistory record in the database   
- Purchase history displayed on the customer account page   
  
---  
  
## 1.24 Manage User Function   
\*\*Function ID:\*\* FR-24   
\*\*Description:\*\* Administrators can manage user accounts, including creating, updating, or deactivating customer and admin accounts.   
\*\*Input:\*\*   
- User ID (string)   
- Username (string)   
- Email (string)   
- Password (string)   
- Status (string)   
- Role (string)   
\*\*Output:\*\*   
- Updated or created user account in the database   
- Confirmation message displayed to the admin   
- User account status reflected in the interface   
  
---  
  
These functional requirements cover the core capabilities of the \*\*GAMMA-J Web Store\*\* system as defined by the use cases and data model. Each function is designed to be feasible, consistent, and aligned with the system’s goals of providing secure, intuitive, and extensible online store management.

# External Description

# Chapter 2: External Interfaces  
  
This chapter defines the external interfaces of the \*\*GAMMA-J Web Store\*\* system. These interfaces represent how the system interacts with users, other software systems, hardware devices, and communication networks to fulfill its functional requirements.  
  
---  
  
## 2.1 User Interface  
  
The user interface is the primary way users interact with the system. It includes web pages, forms, dashboards, and confirmation messages displayed to customers and administrators.  
  
### 2.1.1 Customer Registration Page  
- \*\*Description:\*\* A form where customers enter their personal information (name, email, password, optional address).  
- \*\*Inputs:\*\* Name, Email, Password, Address (optional)  
- \*\*Outputs:\*\* Confirmation message, redirect to login page, and a new account created in the database.  
- \*\*Interaction Method:\*\* Web-based HTML/JavaScript form submission via HTTP POST request.  
  
### 2.1.2 Customer Login Page  
- \*\*Description:\*\* A form where customers enter their email and password to log in.  
- \*\*Inputs:\*\* Email, Password  
- \*\*Outputs:\*\* Redirect to homepage or dashboard, session created, login record logged.  
- \*\*Interaction Method:\*\* Web-based HTML/JavaScript form submission via HTTP POST request.  
  
### 2.1.3 Admin Login Page  
- \*\*Description:\*\* A form where administrators enter their username and password to access the admin interface.  
- \*\*Inputs:\*\* Username, Password  
- \*\*Outputs:\*\* Redirect to admin dashboard, session created, login record logged.  
- \*\*Interaction Method:\*\* Web-based HTML/JavaScript form submission via HTTP POST request.  
  
### 2.1.4 Product Detail View  
- \*\*Description:\*\* Displays detailed product information including name, description, price, specifications, and images.  
- \*\*Inputs:\*\* Product ID, User ID  
- \*\*Outputs:\*\* Displayed product details, option to add to cart.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.5 Cart Summary Page  
- \*\*Description:\*\* Displays a summary of items in the shopping cart, including total cost and item count.  
- \*\*Inputs:\*\* Cart ID, Cart Item list  
- \*\*Outputs:\*\* Cart summary with item count and total cost, error if cart is empty.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.6 Order Confirmation Page  
- \*\*Description:\*\* Displays order confirmation details after checkout, including items, total cost, and shipping address.  
- \*\*Inputs:\*\* Order ID, Customer ID  
- \*\*Outputs:\*\* Order confirmation display, error if no orders found.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.7 Purchase History Page  
- \*\*Description:\*\* Displays a list of completed orders for the customer.  
- \*\*Inputs:\*\* Customer ID  
- \*\*Outputs:\*\* List of past orders with key details (order date, total cost, status), detailed view of selected order.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.8 Admin Dashboard  
- \*\*Description:\*\* A centralized interface for administrators to manage products, inventory, users, and plugins.  
- \*\*Inputs:\*\* Admin ID, Session ID  
- \*\*Outputs:\*\* Access to various administrative functions, confirmation messages, and logs.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.9 Plugin Management Interface  
- \*\*Description:\*\* Allows administrators to install, update, or remove plugins.  
- \*\*Inputs:\*\* Plugin ID, Action (install/update/remove), Plugin file/repository link  
- \*\*Outputs:\*\* Updated plugin status, confirmation message.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.10 Plugin Documentation Viewer  
- \*\*Description:\*\* Displays documentation for a specific plugin.  
- \*\*Inputs:\*\* Plugin ID, Admin ID  
- \*\*Outputs:\*\* Structured documentation content, search/navigation options, error if documentation is missing.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
---  
  
## 2.2 Hardware Interface  
  
Currently, the GAMMA-J Web Store does not interact directly with external hardware devices. Therefore, this section is \*\*not applicable\*\*.  
  
---  
  
## 2.3 Software Interface  
  
The system interacts with various software components and external services through APIs, databases, and internal modules.  
  
### 2.3.1 Database Interface  
- \*\*Description:\*\* The system communicates with a relational database to store and retrieve user data, product information, inventory levels, orders, and logs.  
- \*\*Input:\*\* SQL queries, JSON payloads, etc.  
- \*\*Output:\*\* Data retrieved from the database, updates made to the database.  
- \*\*Interaction Method:\*\* RESTful API or direct database connection using ORM tools.  
- \*\*Data Sources:\*\*  
 - `users` table  
 - `products` table  
 - `orders` table  
 - `cart\_items` table  
 - `purchase\_history` table  
 - `plugins` table  
 - `logs` table  
  
### 2.3.2 Email Service Interface  
- \*\*Description:\*\* The system uses an email service to send confirmation emails to customers.  
- \*\*Input:\*\* Email content, recipient address, subject, order details  
- \*\*Output:\*\* Sent email, delivery status logged  
- \*\*Interaction Method:\*\* SMTP protocol or third-party email API (e.g., SendGrid, Amazon SES)  
- \*\*Data Sources:\*\*  
 - Customer email address  
 - Order details (items, total cost, shipping address)  
  
### 2.3.3 Payment Gateway Interface  
- \*\*Description:\*\* The system integrates with a payment gateway to process transactions.  
- \*\*Input:\*\* Payment method, amount, customer details, order information  
- \*\*Output:\*\* Payment confirmation status, transaction ID  
- \*\*Interaction Method:\*\* REST API or SDK provided by the payment provider (e.g., Stripe, PayPal)  
- \*\*Data Sources:\*\*  
 - Payment method (credit card, digital wallet, etc.)  
 - Customer ID  
 - Cart ID  
 - Order details  
  
### 2.3.4 Inventory Management System  
- \*\*Description:\*\* The system interacts with an internal inventory management module to update stock levels.  
- \*\*Input:\*\* Product ID, New inventory quantity  
- \*\*Output:\*\* Updated inventory records  
- \*\*Interaction Method:\*\* Internal API call within the system  
- \*\*Data Sources:\*\*  
 - `inventory` table  
 - `products` table  
  
### 2.3.5 Logging System  
- \*\*Description:\*\* The system logs all user actions, such as login attempts, order creation, and plugin management.  
- \*\*Input:\*\* Log entry (action type, timestamp, user ID, details)  
- \*\*Output:\*\* Stored log records in the system database  
- \*\*Interaction Method:\*\* Internal logging framework or external logging service  
- \*\*Data Sources:\*\*  
 - `logs` table  
 - User ID  
 - Session ID  
  
---  
  
## 2.4 Communication Interface  
  
The system communicates with external entities via network protocols and messaging systems.  
  
### 2.4.1 Email Communication Interface  
- \*\*Description:\*\* The system sends order confirmation emails to customers.  
- \*\*Input:\*\* Order details, customer email, subject line  
- \*\*Output:\*\* Email sent, delivery status recorded  
- \*\*Interaction Method:\*\* SMTP protocol or third-party email API  
- \*\*Data Sources:\*\*  
 - Customer email address  
 - Order details (items, total cost, shipping address)  
  
### 2.4.2 Web Browser Communication Interface  
- \*\*Description:\*\* The system communicates with web browsers via HTTP/HTTPS protocols to serve web pages and handle user interactions.  
- \*\*Input:\*\* HTTP requests (GET, POST, PUT, DELETE)  
- \*\*Output:\*\* HTTP responses containing HTML, JSON, or XML data  
- \*\*Interaction Method:\*\* RESTful API endpoints  
- \*\*Data Sources:\*\*  
 - User input data (form submissions)  
 - Session tokens  
 - Product catalog  
 - Order history  
  
### 2.4.3 API Communication Interface  
- \*\*Description:\*\* The system exposes and consumes RESTful APIs for internal and external integration.  
- \*\*Input:\*\* API requests with query parameters, headers, and body data  
- \*\*Output:\*\* API responses with status codes, data payloads, or error messages  
- \*\*Interaction Method:\*\* HTTP/HTTPS with JSON/XML payload  
- \*\*Data Sources:\*\*  
 - User accounts  
 - Product catalog  
 - Orders  
 - Logs  
 - Plugins  
  
---  
  
## Summary of External Interfaces  
  
| Interface Type | Description | Interaction Method | Data Sources Involved |  
|----------------------|-----------------------------------------------------------------------------|----------------------------|----------------------------------------------------------------------------------------|  
| User Interface | All user-facing screens and forms | Web-based HTML/JS | Users, Products, Orders, Cart, Logs, Purchase History, Plugins |  
| Software Interface | Integration with database, payment gateways, inventory, and logging systems | REST API, SQL, ORM | Users, Products, Orders, Cart, Logs, Purchase History, Inventory, Plugins |  
| Communication Interface | Email notifications, web browsing, and API calls | SMTP, HTTP/HTTPS, REST | Users, Products, Orders, Cart, Logs, Purchase History, Inventory, Plugins |  
  
This chapter ensures that all external interactions are clearly defined, consistent with the functional requirements, and ready for implementation.