QUEZON CITY UNIVERSITY
COLLEGE OF COMPUTER STUDIES

Product Catalog Management System

Lina, Jhan Russel C.

Lingcopines Jr., Rodel L.

Listangco, Frank John E.

Lobrio, Ivahn Mika F.

Lopez, Jefferson C.

I. DATABASE DESIGN

I. Data Flow Diagram

Administrator Side

Administrators have full control over the product catalog. They can create, update, and delete product information, ensuring the catalog is accurate and up-to-date. This includes managing product details such as name, description, supplier, price, and images. Admins also manage product categories, allowing for organized browsing and searching. To maintain accurate stock information, administrators update inventory levels, reflecting new products, sales, and any necessary adjustments. Before any data is stored, the system automatically validates and sanitizes it, preventing errors and security risks. This ensures data integrity and a smooth user experience.

Admin Activities:

- **Product Management:**
  - The Admin creates, updates, and deletes product data (1.0).
  - Before storage, the system validates and sanitizes this data (4.0) to ensure data integrity and security.
  - Validated product data is then stored in the Product Data Store.
- **Category Management:**
  - The Admin creates, updates, and deletes category data (2.0), which is also validated and sanitized (4.0) before being stored in the Category Data Store.
- **Inventory Management:**
  - The Admin updates inventory data (3.0), subject to validation and sanitization (4.0), and the updated information is stored in the Inventory Data Store.

Customer Side

Users can easily explore the product catalog through browsing and searching. They can browse products by category or use the search function to find specific items. To refine their search, users can apply filters based on criteria like price range, suppliers (brand) or other relevant attributes. The system then presents them with a clear and organized list of products matching their criteria, allowing for efficient product discovery and comparison.

User Activities:

- **Product Browsing and Searching:**
  - Users provide search criteria and filters (5.0) to find specific products.

- The system retrieves relevant product data from the Product Data Store and category information from the Category Data Store based on the user's request.
- The system then presents the product information, including search results and filtered lists, to the use

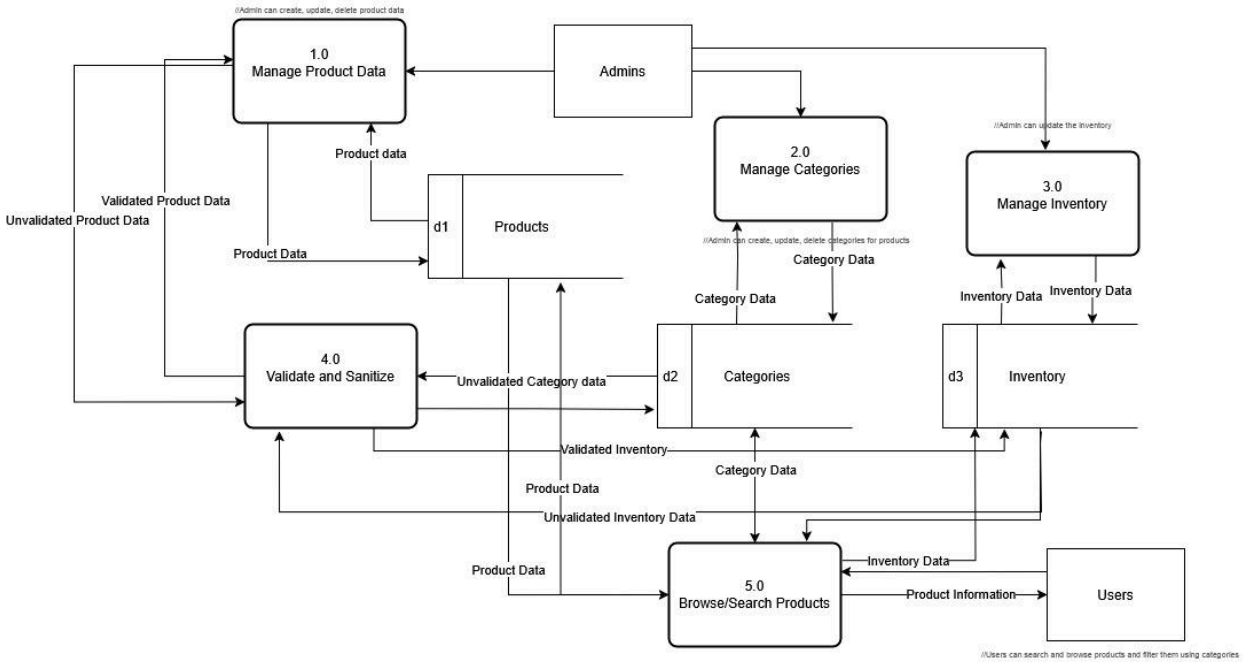


DIAGRAM LINK:
https://drive.google.com/file/d/1v1W05joCodiU5qn2e44cFZ8ehW8lQs6v/view?usp=sharing

II. Data Dictionary

Product Table

Column Name	Data Type	Constraints	Description
product_id	varchar (10)	PK, NOT NULL	The unique identifier for each product
category_id	varchar(20)	FK	Foreign key referencing to the product's category
supplier_id	varchar(15)	FK	Foreign key referencing to the product's supplier
product_name	varchar(20)	NOT NULL	Name of the product
product_img	BLOB		Binary data for the product image
price	decimal(6, 2)		Price of the product
description	Text		Description of the product
created_at	datetime		Date and time when the product was created
updated_at	datetime		Date and time when the product was last updated

Category Table

Column Name	Data Type	Constraints	Description
category_id	varchar(20)	FK	Foreign key referencing to the product's category
category_name	varchar(20)	NOT NULL	Name of the category
description	Text		Detailed description of the category
created_at	datetime		Date and time when the category was created

Inventory Table

Column Name	Data Type	Constraints	Description
inventory_id	varchar(10)	PK, NOT NULL	Unique identifier for each inventory record
product_id	varchar(20)	FK	Foreign Key referencing product table
description	Text		Detailed description of the inventory
stock_quantity	int(5)		The quantity of a certain product
last_updated	datetime		Date and time when the inventory was last updated

Admin Table

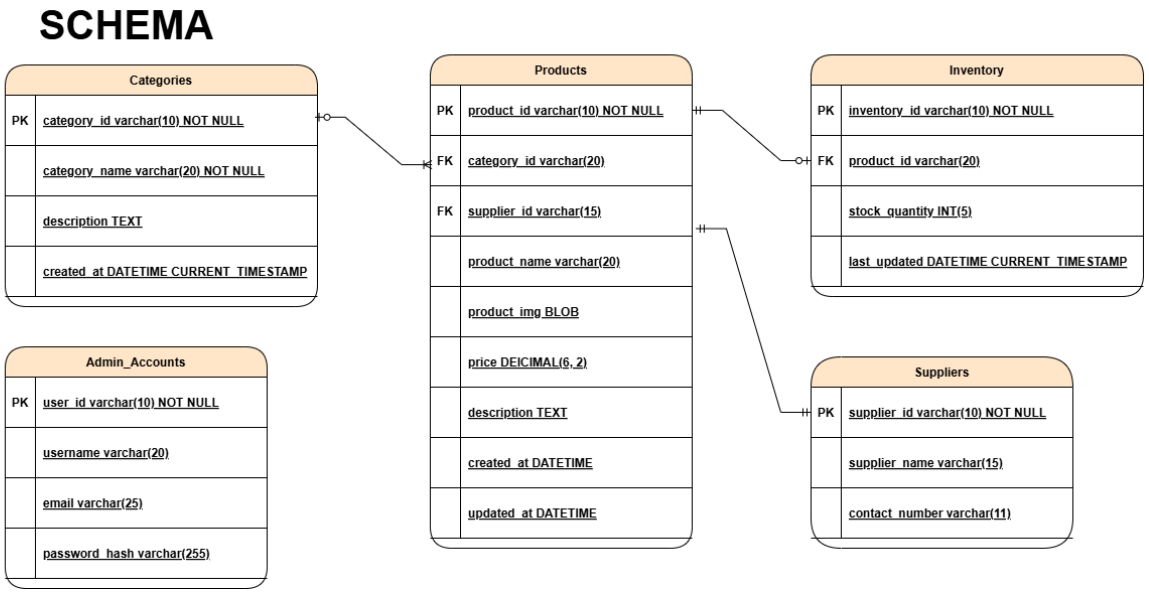
Column Name	Data Type	Constraints	Description
user_id	varchar(10)	PK, NOT NULL	Unique identifier for each admin user
username	varchar(20)	NOT NULL	Username of the admin
email	varchar(25)	UNIQUE	Email address of the admin

stock_quantity	int(5)		Date and time when the product was created
password_hash	varchar(255)		Hashed password of the admin

Supplier Table

Column Name	Data Type	Constraints	Description
supplier_id	varchar(10)	PK, NOT NULL	Unique identifier for each supplier
supplier_name	varchar(15)	NOT NULL	Name of the supplier
contact_number	varchar(11)		Contact number of the supplier

III. Schema



1. Categories Table

- **Table Name:** `Categories`
- **Purpose:** Stores information about different product categories.
- **Columns:**
  1. `category_id` **VARCHAR(10) NOT NULL** – Primary Key, a unique identifier for each category.
  2. `category_name` **VARCHAR(20) NOT NULL** – The name of the category (e.g., "Electronics").
  3. `description` **TEXT** – A text description providing more details about the category.
  4. `created_at` **DATETIME DEFAULT CURRENT_TIMESTAMP** – Timestamp of when the category record was created.

2. Products Table

- **Table Name:** `Products`
- **Purpose:** Stores product details, including images, pricing, and references to categories and suppliers.
- **Columns:**
  1. `product_id` **VARCHAR(10) NOT NULL** – Primary Key, a unique product identifier (e.g., "PROD001").
  2. `category_id` **VARCHAR(20)** – Foreign Key referencing `Categories(category_id)`, linking the product to its category.
  3. `supplier_id` **VARCHAR(10)** – Foreign Key referencing `Suppliers(supplier_id)`, linking the product to its supplier.
  4. `product_name` **VARCHAR(20) NOT NULL** – The name of the product (e.g., "iPhone 14").
  5. `product_image` **BLOB** – Stores binary data (images, etc.).
  6. `price` **DECIMAL(6,2)** – The product's price, allowing up to 6 digits and 2 decimal places (e.g., `999.99`).
  7. `description` **TEXT** – A text field containing product details.
  8. `created_at` **DATETIME** – Timestamp of when the product record was created.
  9. `updated_at` **DATETIME** – Timestamp of the most recent update to the product record.

3. Inventory Table

- **Table Name:** `Inventory`
- **Purpose:** Manages stock levels for each product.
- **Columns:**
  1. `inventory_id` **VARCHAR(10) NOT NULL** – Primary Key, a unique identifier for each inventory record.

2. `product_id` **VARCHAR(20) NOT NULL** – Foreign Key referencing `Products(product_id)`, linking the stock record to a specific product.
3. `stock_quantity` **INT(5)** – The current number of items in stock.
4. `last_updated` **DATETIME DEFAULT CURRENT_TIMESTAMP** – Timestamp indicating the last time the stock was updated.

4. Suppliers Table

- **Table Name:** `Suppliers`
- **Purpose:** Maintains information about suppliers who provide the products.
- **Columns:**
  1. `supplier_id` **VARCHAR(10) NOT NULL** – Primary Key, a unique identifier for each supplier.
  2. `supplier_name` **VARCHAR(15) NOT NULL** – The supplier's name (e.g., "Tech Distributors").
  3. `contact_number` **VARCHAR(11) NOT NULL** – The supplier's contact phone number.

5. Admin_Accounts Table

- **Table Name:** `Admin_Accounts`
- **Purpose:** Stores administrative user details for system access and management.
- **Columns:**
  1. `user_id` **VARCHAR(10) NOT NULL** – Primary Key, a unique identifier for each admin user.
  2. `username` **VARCHAR(20) NOT NULL** – The admin's username (e.g., "adminUser").
  3. `email` **VARCHAR(25) NOT NULL** – The admin's email address (e.g., "admin@example.com").
  4. `password_hash` **VARCHAR(255)** – A hashed password for secure authentication (e.g., bcrypt hash).

Relationships and Key Points

1. **Categories ↔ Products**
    - `category_id` in **Products** references `category_id` in **Categories**.
    - This establishes a **one-to-many** relationship (one category can have many products).
2. **Suppliers ↔ Products**
    - `supplier_id` in **Products** references `supplier_id` in **Suppliers**.
    - Also a **one-to-many** relationship (one supplier can supply many products).
3. **Products ↔ Inventory**
    - `product_id` in **Inventory** references `product_id` in **Products**.
    - This is a **one-to-one or one-to-many** relationship depending on whether each product can have multiple inventory records or just one. Typically, it's **one-to-one** if you track a single stock entry per product.
4. **Admin_Accounts**
    - Independent table that holds admin credentials for system management.
5. **BLOB Data (Product Images)**
    - `product_image` in **Products** can store binary data such as images or documents.
    - Ensure you handle **server configurations** (e.g., `max_allowed_packet` in MySQL) for large files.

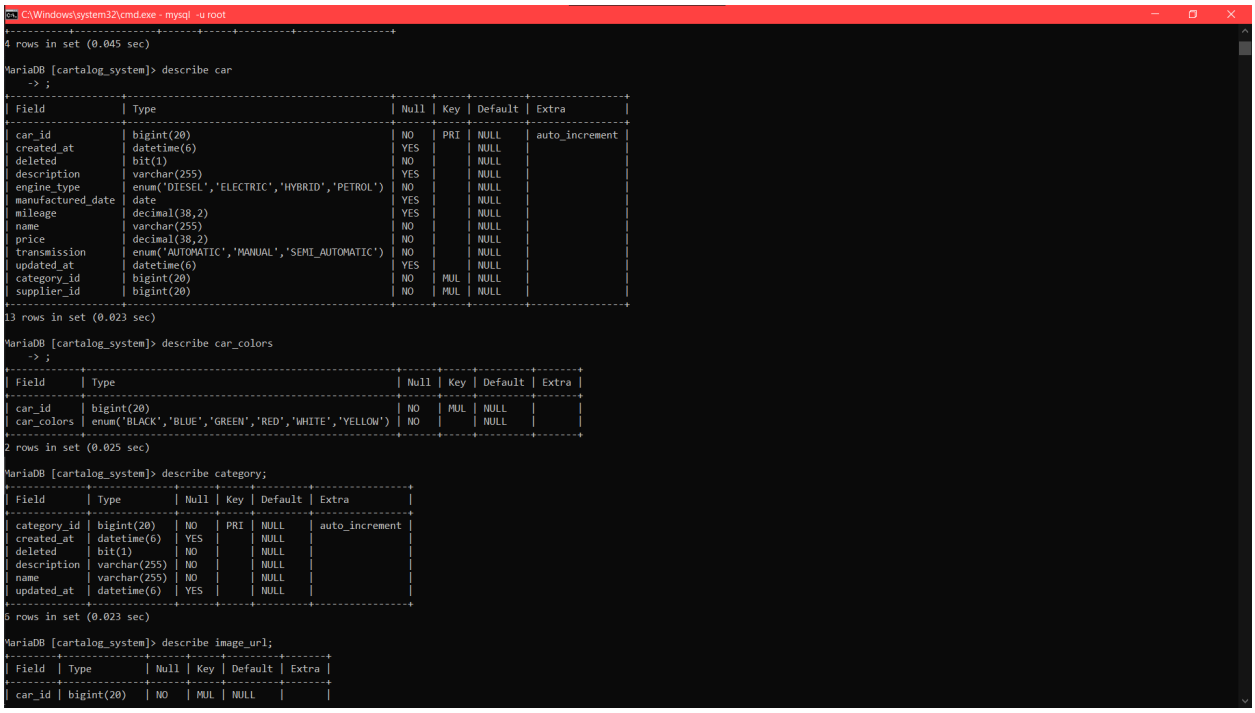
II. PROGRAMMING LANGUAGE DESIGN

COMPONENT	SPECIFICATIONS	DESCRIPTION
Operating System	• Project Catalog Management System	• Project Catalog Management System is a software tool or platform that helps organizations organize, manage and distribute project information and sources.
Programming Languages Used	• Java • HTML	• Java is an object oriented programming language. • HTML is a foundational language for structuring web pages.It defines the content and basic structure of a webpage, including elements like headings, images, and links.
Backend	• Mysql • Java • Postman • SpringBoot	• Mysql is an open source relational database management system (RDMS) that uses sql to manage and store data. • Java is a widely used, general purpose, object oriented programming language and software platform. • Postman is an all in one API platform used for building, testing , and managing APIs. • SpringBoot is a java based framework designed to simplify the development of stand alone, production ready Spring applications.

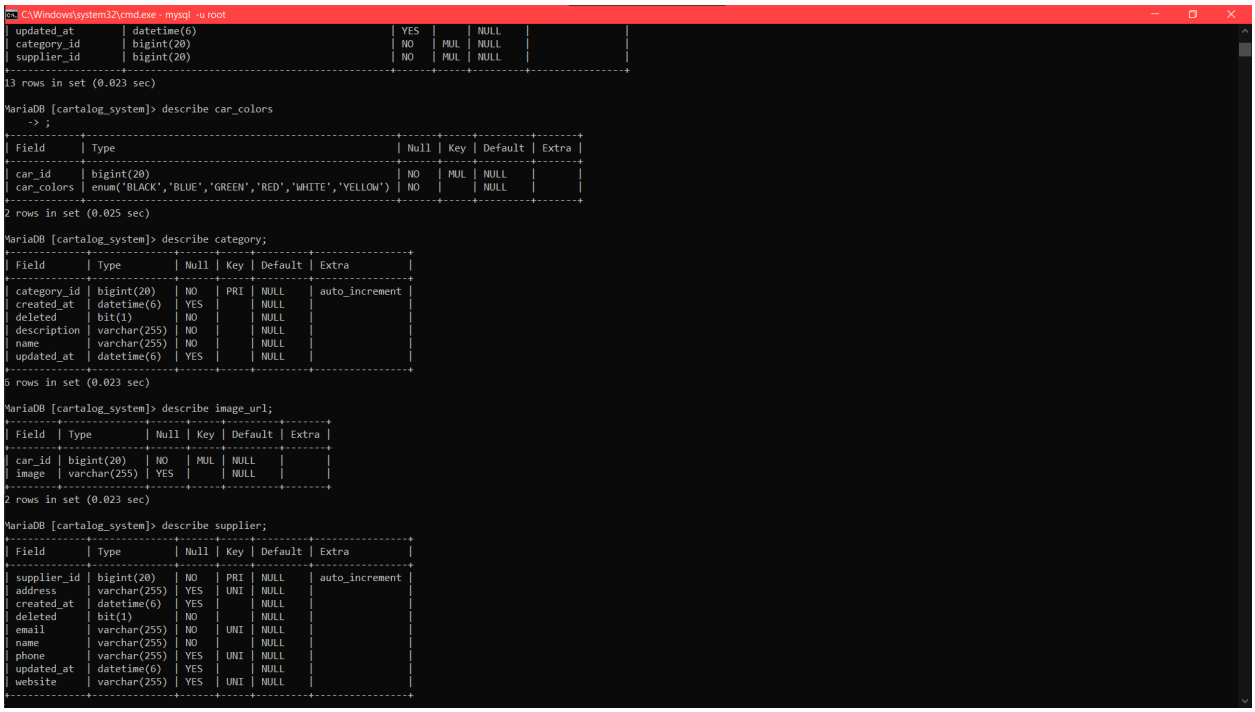
Frontend	• React • Typescript • Tailwind	• React is a free and open source front end java script library that aims to make building a user interface based on components more seamless. • Typescript is a free and open source high level programming language that adds static typing with optional type annotations to Javascript. • Tailwind CSS utility first CSS framework that provides predefined classes for styling elements directly in HTML.

III. TASK IN PROJECT

1. **Design a schema for products, categories, and manufacturers.**

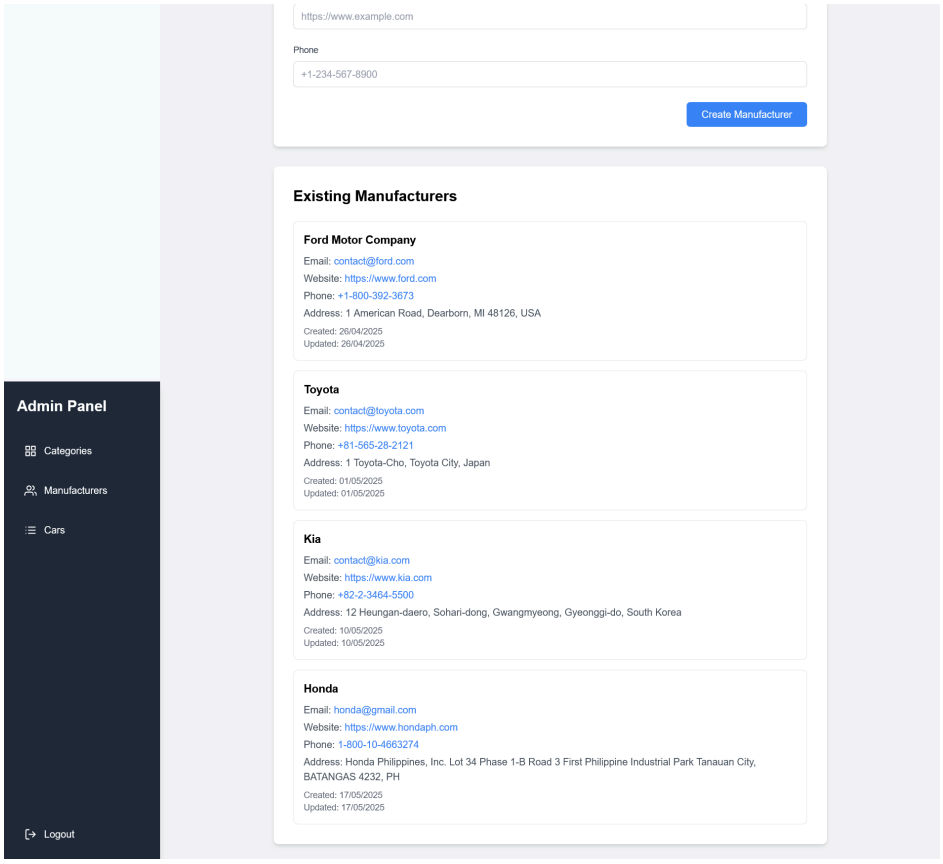


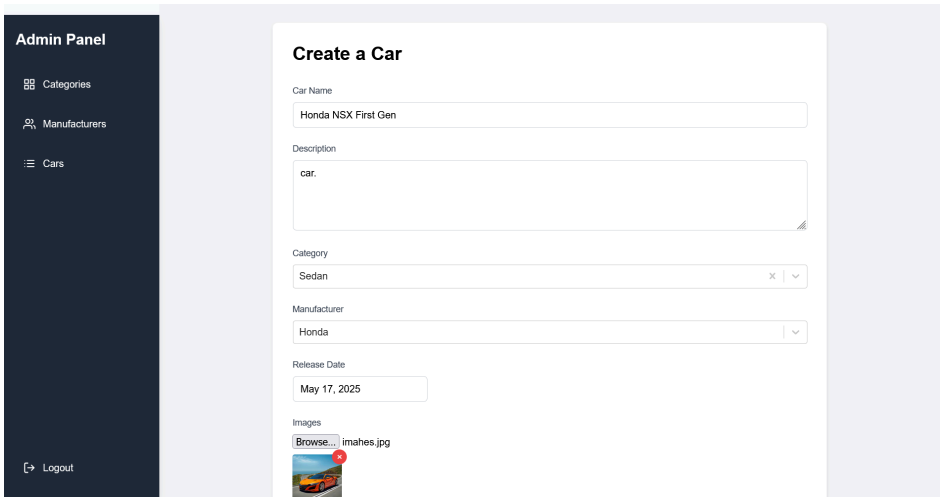
```
C:\Windows\system32\cmd.exe - mysql -u root
+----------+--------------+------+-----+---------+----------------+
4 rows in set (0.045 sec)

MariaDB [cartalog_system]> describe car
    -> ;
+------------------+-------------------------------------------------+------+-----+---------+----------------+
| Field            | Type                                            | Null | Key | Default | Extra          |
+------------------+-------------------------------------------------+------+-----+---------+----------------+
| car_id           | bigint(20)                                      | NO   | PRI | NULL    | auto_increment |
| created_at       | datetime(6)                                     | YES  |     | NULL    |                |
| deleted          | bit(1)                                          | NO   |     | NULL    |                |
| description      | varchar(255)                                    | YES  |     | NULL    |                |
| engine_type      | enum('DIESEL','ELECTRIC','HYBRID','PETROL')     | NO   |     | NULL    |                |
| manufactured_date | date                                           | YES  |     | NULL    |                |
| mileage          | decimal(38,2)                                   | YES  |     | NULL    |                |
| name             | varchar(255)                                    | NO   |     | NULL    |                |
| price            | decimal(38,2)                                   | NO   |     | NULL    |                |
| transmission     | enum('AUTOMATIC','MANUAL','SEMI_AUTOMATIC')     | NO   |     | NULL    |                |
| updated_at       | datetime(6)                                     | YES  |     | NULL    |                |
| category_id      | bigint(20)                                      | NO   | MUL | NULL    |                |
| supplier_id      | bigint(20)                                      | NO   | MUL | NULL    |                |
+------------------+-------------------------------------------------+------+-----+---------+----------------+
13 rows in set (0.023 sec)

MariaDB [cartalog_system]> describe car_colors
    -> ;
+------------+-----------------------------------------------------+------+-----+---------+-------+
| Field      | Type                                                | Null | Key | Default | Extra |
+------------+-----------------------------------------------------+------+-----+---------+-------+
| car_id     | bigint(20)                                          | NO   | MUL | NULL    |       |
| car_colors | enum('BLACK','BLUE','GREEN','RED','WHITE','YELLOW') | NO   |     | NULL    |       |
+------------+-----------------------------------------------------+------+-----+---------+-------+
2 rows in set (0.025 sec)

MariaDB [cartalog_system]> describe category;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| category_id | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| created_at  | datetime(6)  | YES  |     | NULL    |                |
| deleted     | bit(1)       | NO   |     | NULL    |                |
| description | varchar(255) | NO   |     | NULL    |                |
| name        | varchar(255) | NO   |     | NULL    |                |
| updated_at  | datetime(6)  | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.023 sec)

MariaDB [cartalog_system]> describe image_url;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| car_id | bigint(20)   | NO   | MUL | NULL    |       |
```

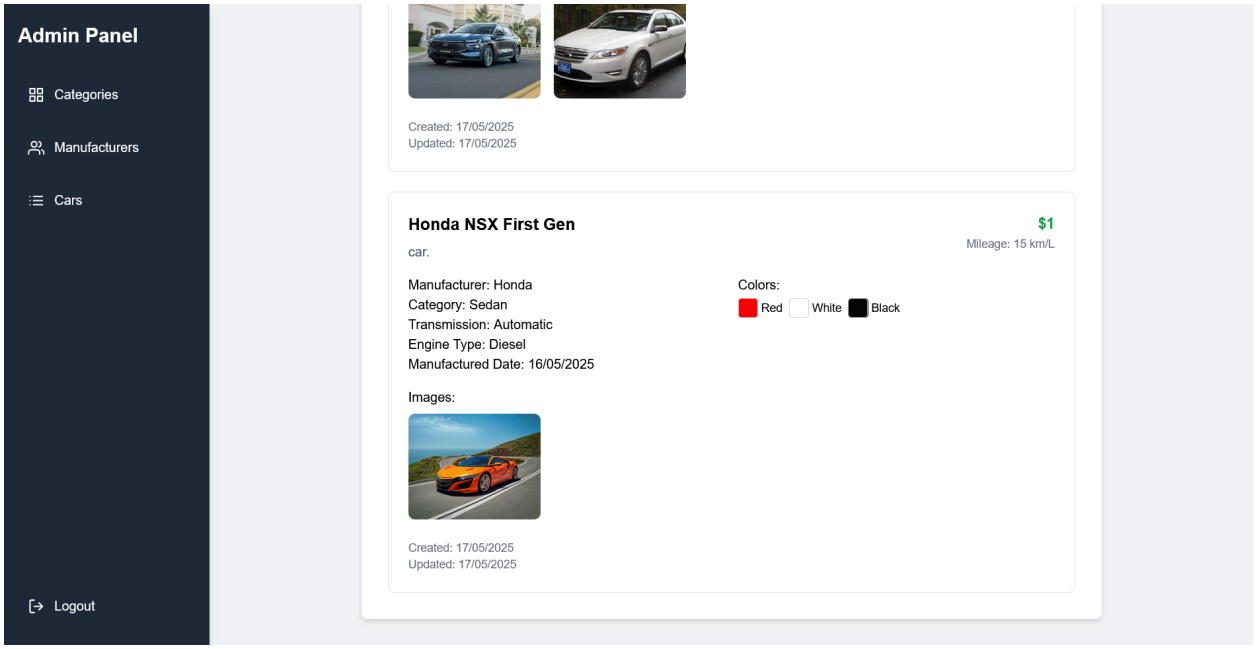


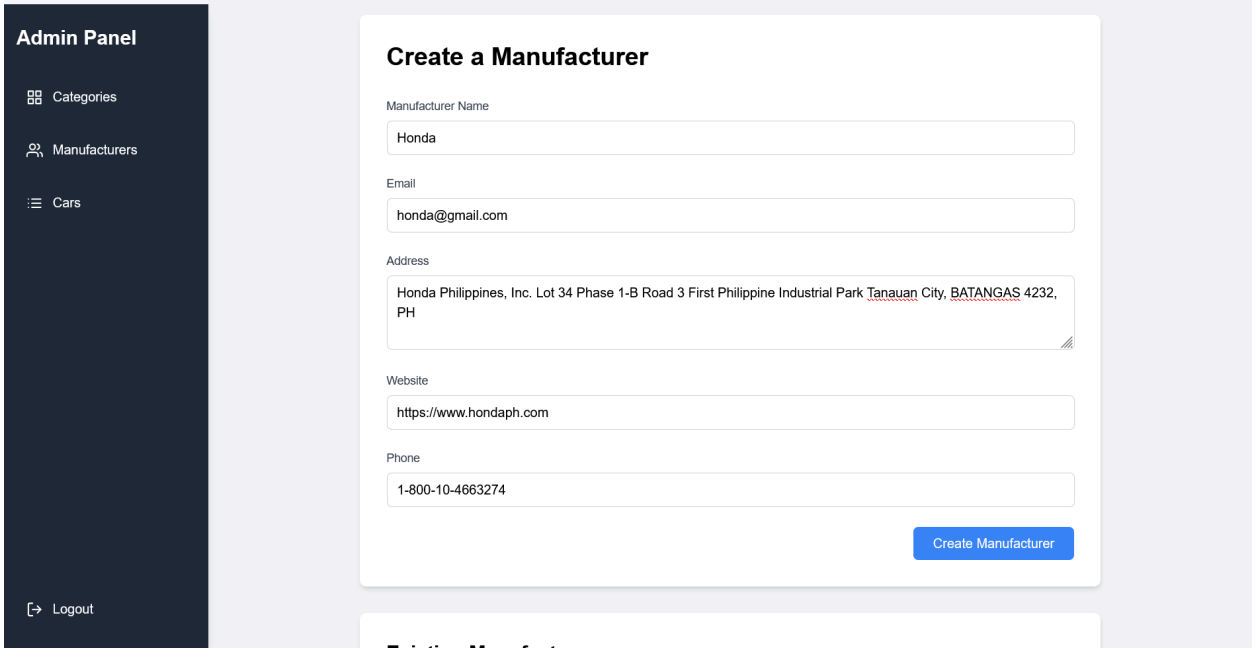
```
C:\Windows\system32\cmd.exe - mysql -u root
| updated_at       | datetime(6)                                     | YES  |     | NULL    |                |
| category_id      | bigint(20)                                      | NO   | MUL | NULL    |                |
| supplier_id      | bigint(20)                                      | NO   | MUL | NULL    |                |
+------------------+-------------------------------------------------+------+-----+---------+----------------+
13 rows in set (0.023 sec)

MariaDB [cartalog_system]> describe car_colors
    -> ;
+------------+-----------------------------------------------------+------+-----+---------+-------+
| Field      | Type                                                | Null | Key | Default | Extra |
+------------+-----------------------------------------------------+------+-----+---------+-------+
| car_id     | bigint(20)                                          | NO   | MUL | NULL    |       |
| car_colors | enum('BLACK','BLUE','GREEN','RED','WHITE','YELLOW') | NO   |     | NULL    |       |
+------------+-----------------------------------------------------+------+-----+---------+-------+
2 rows in set (0.025 sec)

MariaDB [cartalog_system]> describe category;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| category_id | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| created_at  | datetime(6)  | YES  |     | NULL    |                |
| deleted     | bit(1)       | NO   |     | NULL    |                |
| description | varchar(255) | NO   |     | NULL    |                |
| name        | varchar(255) | NO   |     | NULL    |                |
| updated_at  | datetime(6)  | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.023 sec)

MariaDB [cartalog_system]> describe image_url;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| car_id | bigint(20)   | NO   | MUL | NULL    |       |
| image  | varchar(255) | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
2 rows in set (0.023 sec)

MariaDB [cartalog_system]> describe supplier;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| supplier_id | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| address     | varchar(255) | YES  | UNI | NULL    |                |
| created_at  | datetime(6)  | YES  |     | NULL    |                |
| deleted     | bit(1)       | NO   |     | NULL    |                |
| email       | varchar(255) | NO   | UNI | NULL    |                |
| name        | varchar(255) | NO   |     | NULL    |                |
| phone       | varchar(255) | YES  | UNI | NULL    |                |
| updated_at  | datetime(6)  | YES  |     | NULL    |                |
| website     | varchar(255) | YES  | UNI | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
```

2. Implement basic CRUD operations to manage product data.



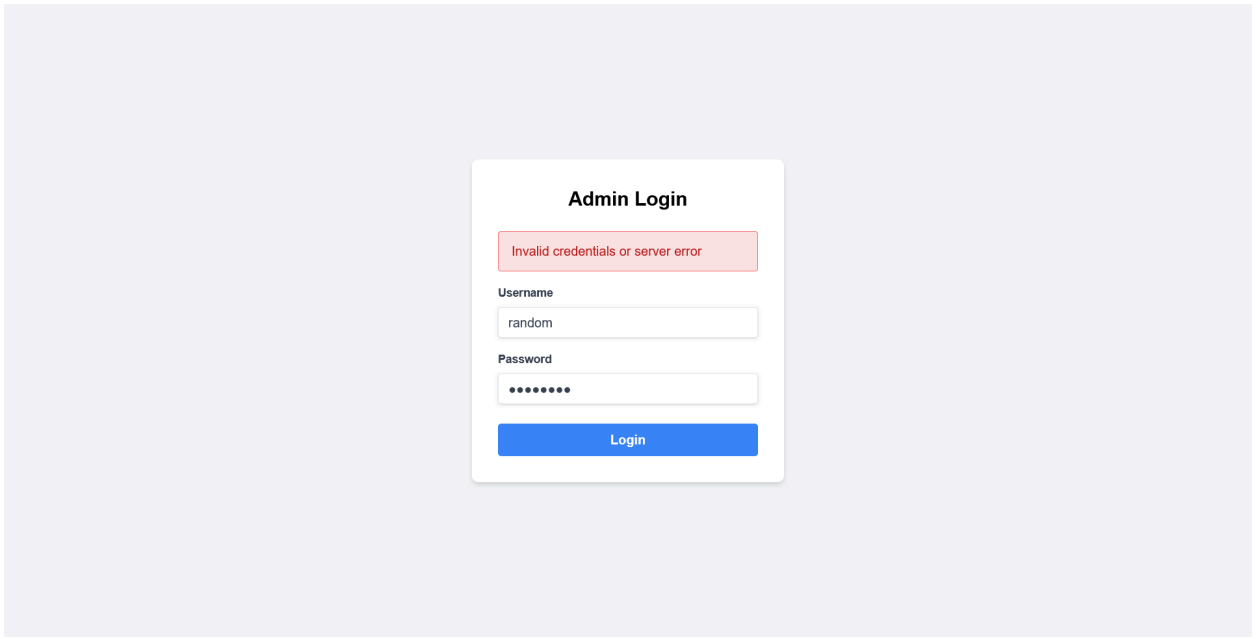


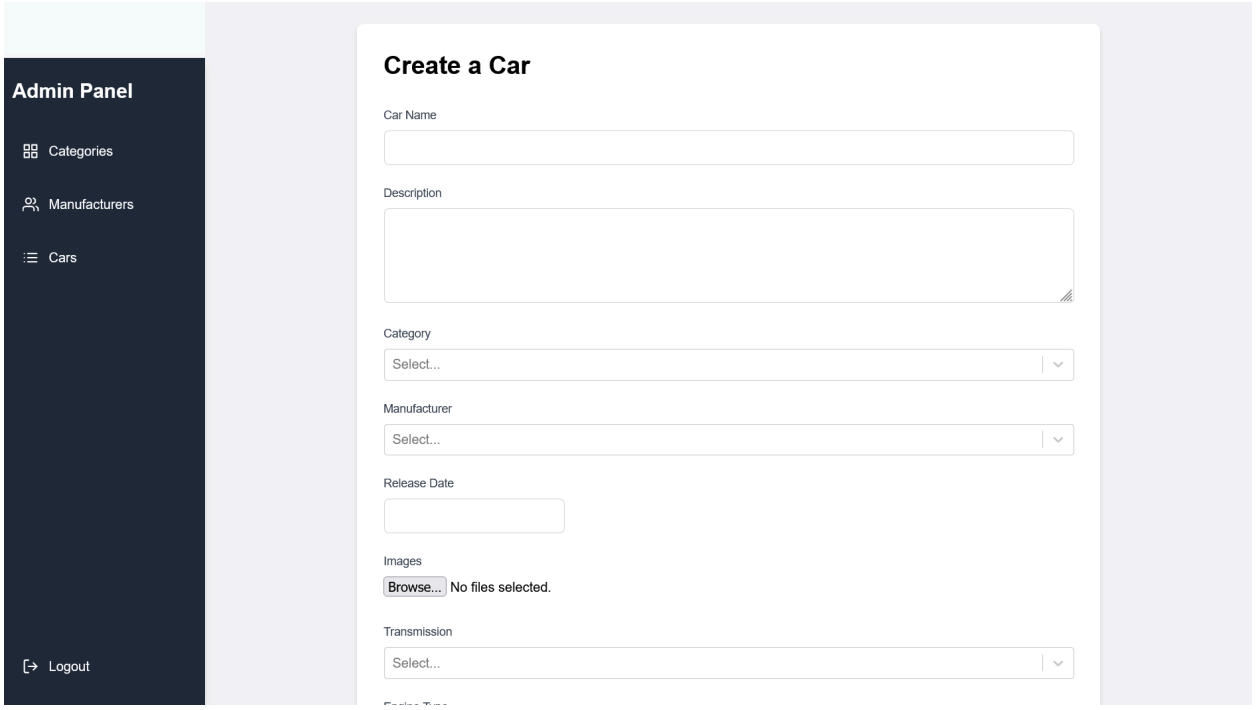


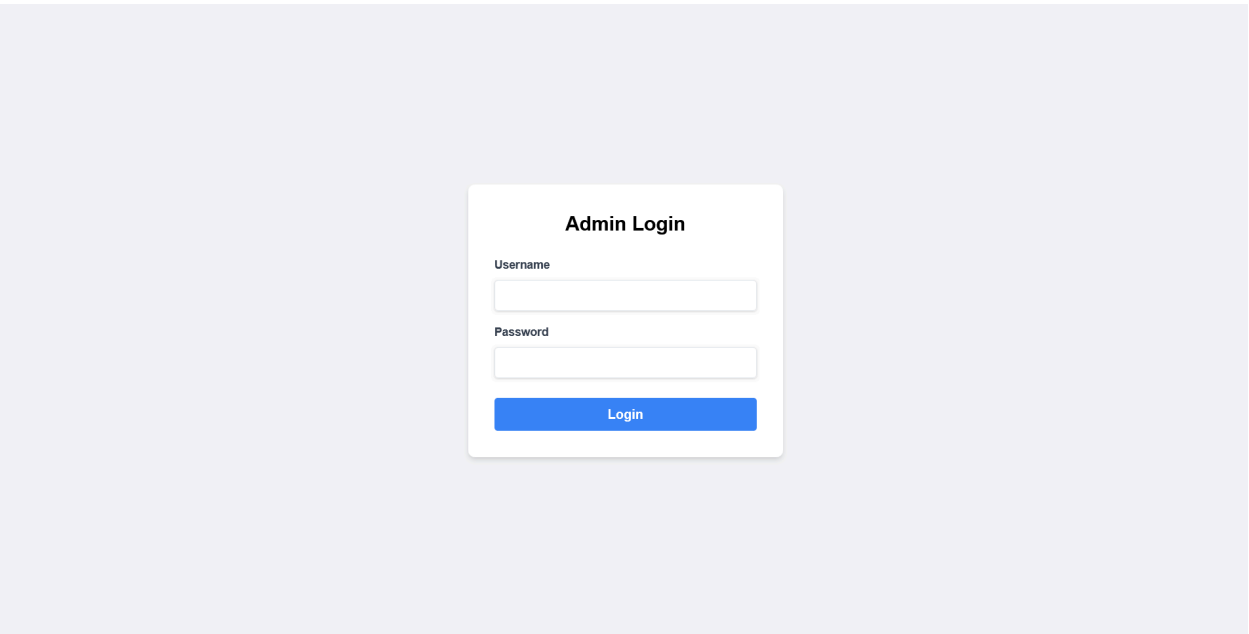


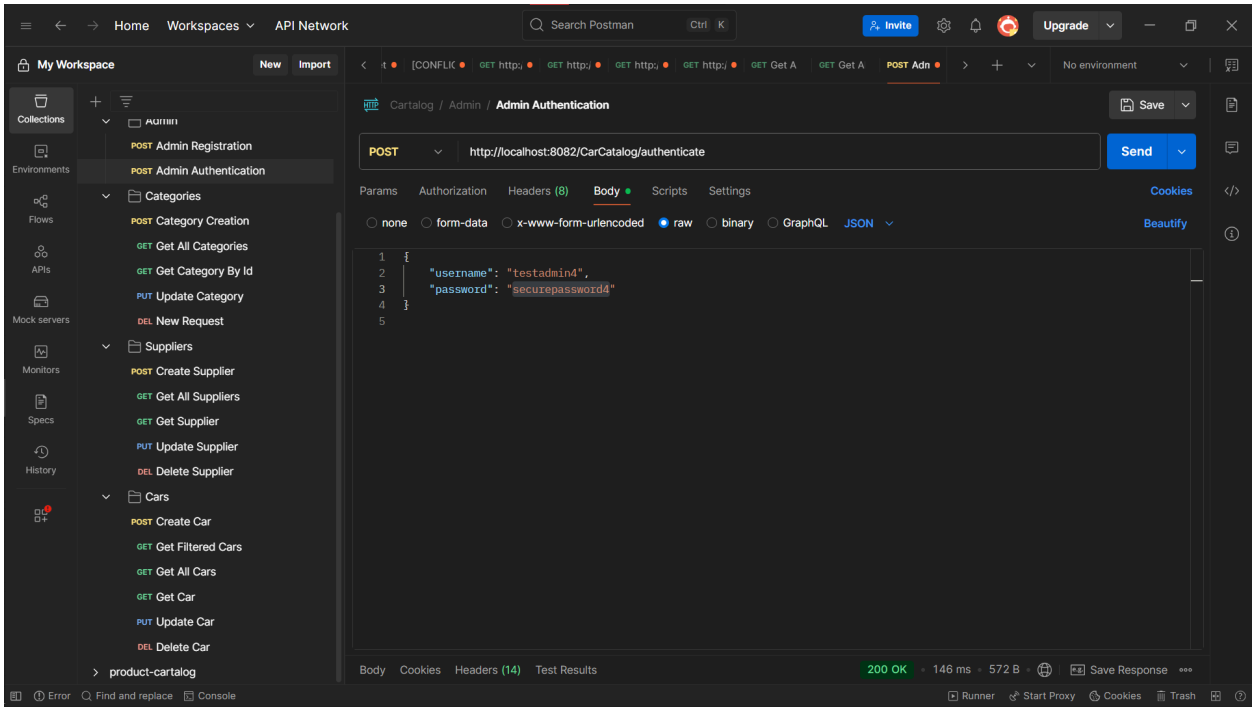
3. **Include input validation and sanitize database queries.**

QUEZON CITY UNIVERSITY
COLLEGE OF COMPUTER STUDIES

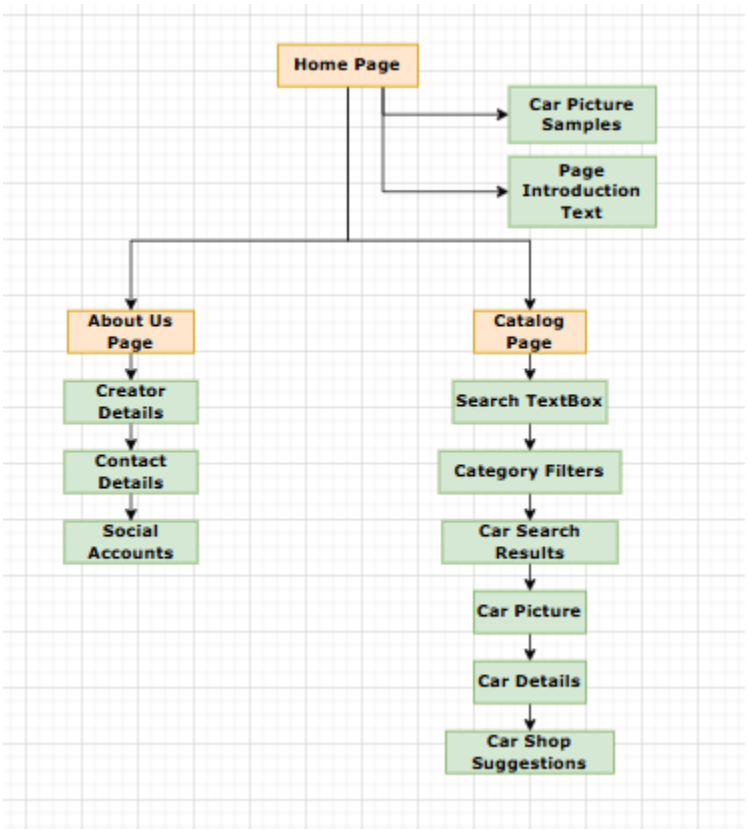





Admin Login
Username
Password
Login


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
Flows
APIs
Mock servers
Monitors
Specs
History
POST Admin Registration
POST Admin Authentication
Categories
POST Category Creation
GET Get All Categories
GET Get Category By Id
PUT Update Category
DEL New Request
Suppliers
POST Create Supplier
GET Get All Suppliers
GET Get Supplier
PUT Update Supplier
DEL Delete Supplier
Cars
POST Create Car
GET Get Filtered Cars
GET Get All Cars
GET Get Car
PUT Update Car
DEL Delete Car
product-cartalog
Cartalog / Admin / Admin Authentication
Save
POST
http://localhost:8082/CarCatalog/authenticate
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"username": "testadmin4",
"password": "securepassword4"
}
Body
Cookies
Headers (14)
Test Results
200 OK
146 ms
572 B
Save Response
Error
Find and replace
Console
Runner
Start Proxy
Cookies
Trash

IV. Sitemap



1. Home Page (Central Hub)

- **Visual Anchors**: "Car Picture Samples" serve as high-fidelity stimuli to orient users visually and reinforce domain context.
- **Introductory Content**: "Page Introduction Text" provides task framing, system affordances, and navigational cues.

2. About Us Page (Provenance & Credibility)

- **Creator Details**: Metadata on the development team emphasizes authorship and expertise.
- **Contact Details**: Multiple touchpoints for support and feedback, reducing user uncertainty.

- **Social Accounts**: Channels for community engagement and real-time updates, fostering trust and follow-through.

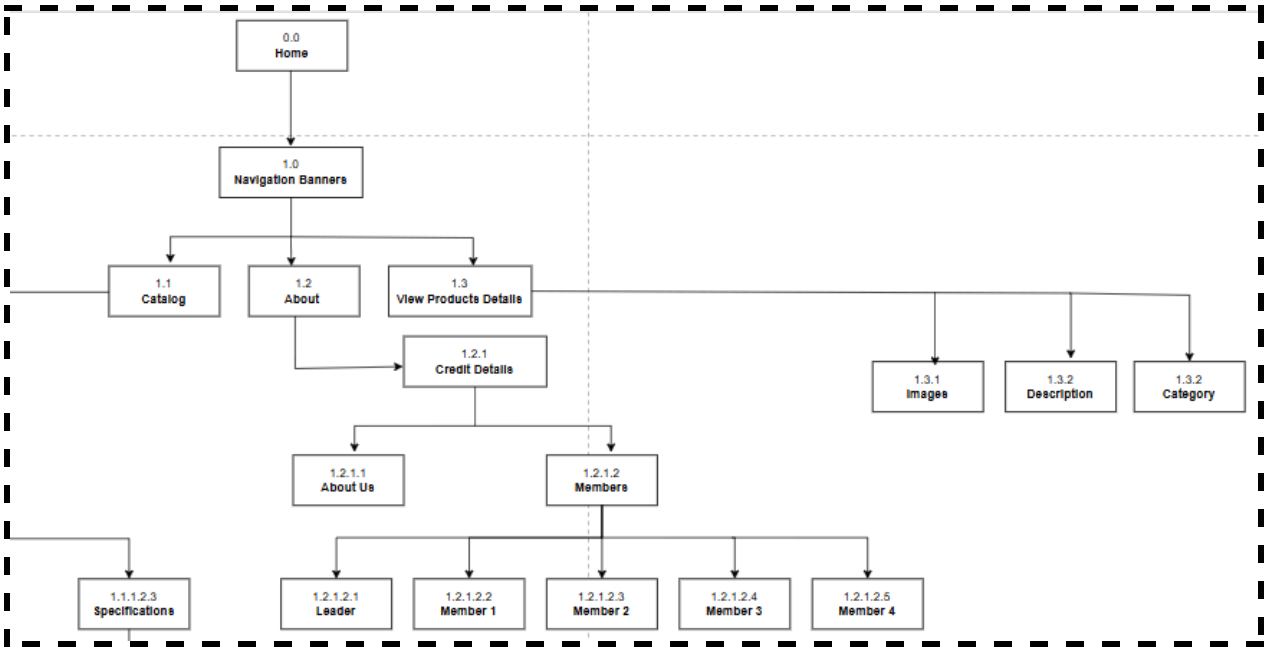
3. Catalog Page (Discovery & Management Core)

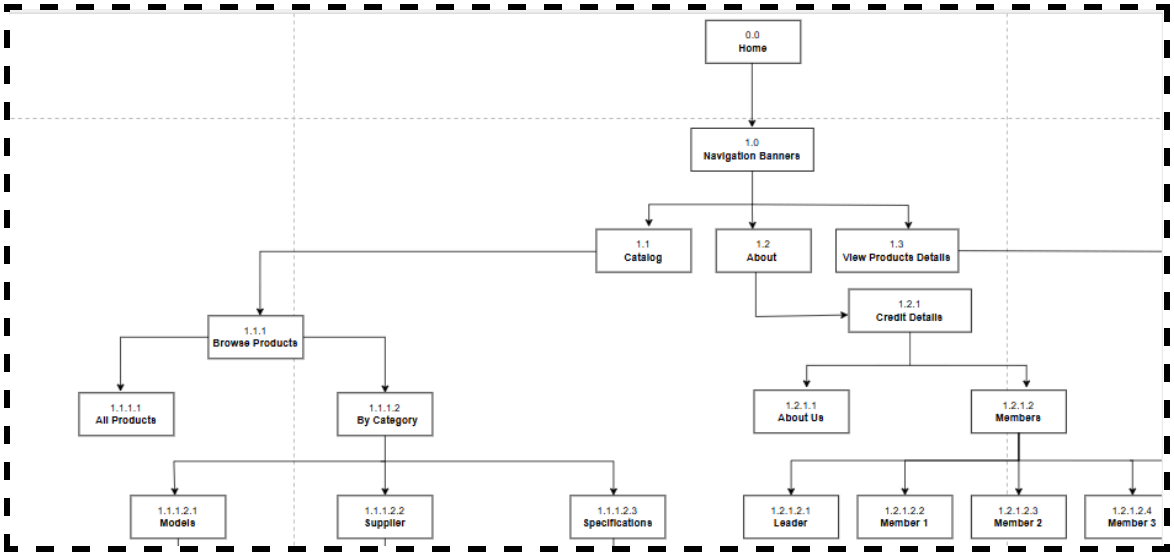
- **Search TextBox**
    - **Function**: Enables keyword-based retrieval across the entire inventory.
    - **Research Implication**: Supports exploratory search behaviors and satisfies both known-item and exploratory queries.
- **Category Filters**
    - **Taxonomy**: Multi-faceted filters (make, model, year, price range, etc.) facilitate drill-down.
    - **Cognitive Load**: Carefully scoped filter sets prevent choice overload while maintaining expressivity.
- **Car Search Results**
    - **Thumbnail Grid**: Visual browse patterns support rapid visual scanning and comparison.
    - **Progressive Disclosure**: Results present minimal metadata (e.g., model name, price) to scaffold attention.
- **Car Detail View**
    - **Car Picture**: High-resolution imagery for visual inspection.
    - **Car Details**: Structured specs (engine, dimensions, features) leverage tabular comprehension.
- **Car Shop Suggestions**

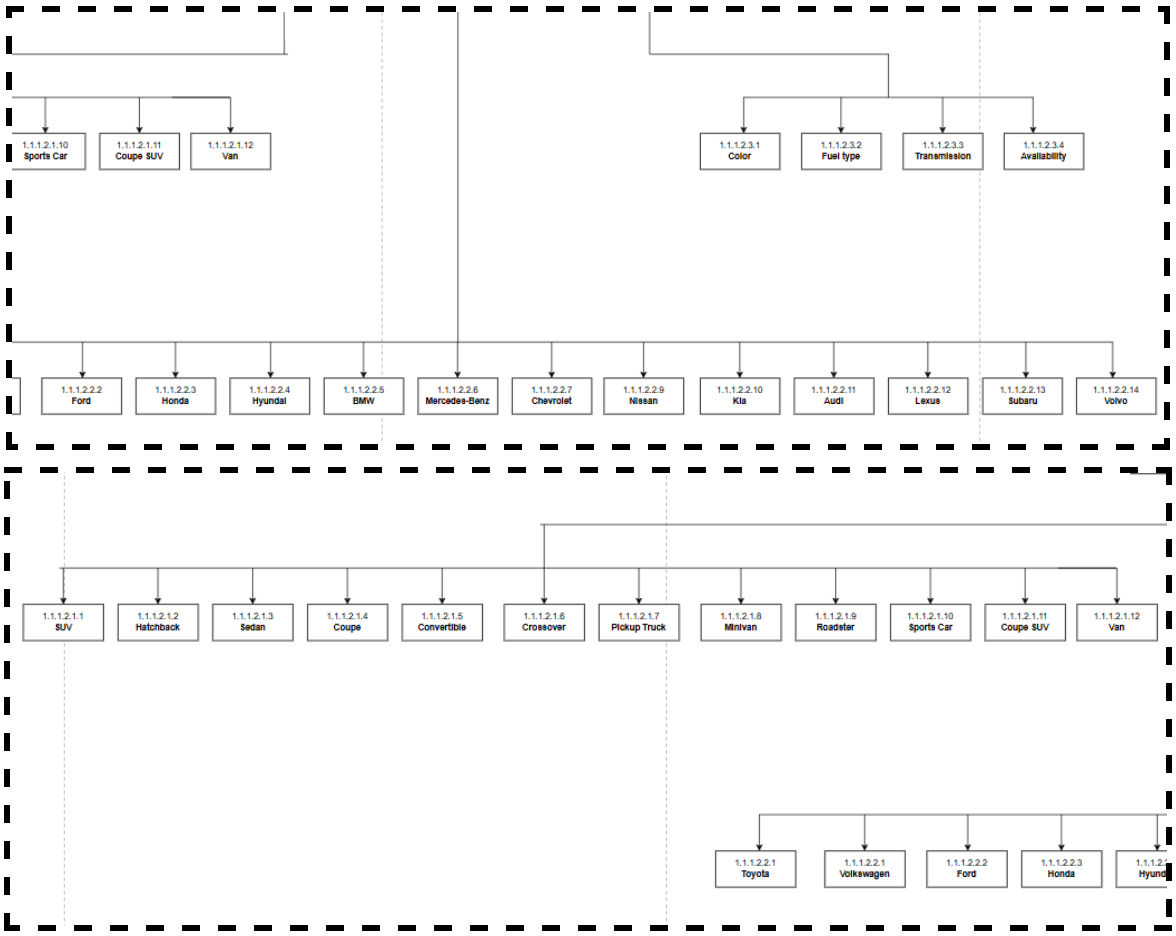
- **Recommendation Engine**: Contextual “related vehicles” to promote serendipitous discovery and cross-sell opportunities.

- **Interface Design**: Inline suggestions reduce navigation friction between detail exploration and further browsing.

V. VTOC







1. Top-Level Navigation

Home (0.0)
A centralized entry point presenting brand identity and global search affordances.
Navigation Banners (1.0)
Persistent menu elements that scaffold rapid access to core system functions.

- **Catalog (1.1)**
  Entry to the primary inventory exploration module.

- **About (1.2)**
  Metadata hub for team provenance and system context.

- **View Product Details (1.3)**
  Direct pathway to in-depth vehicle specifications and media assets.

2. Catalog Substructure (1.1)

A multi-faceted discovery environment enabling both broad and targeted product retrieval.

- **Browse Products (1.1.1)**
  User-driven exploration interface with two navigational pivots:
  - **All Products (1.1.1.1)**
    Unfiltered inventory listing to support serendipitous discovery.
  - **By Category (1.1.1.2)**
    Faceted drill-down experience subdivided into:
    - **Models (1.1.1.2.1)**
      Vehicle body-style taxonomy (SUV, Hatchback, Sedan, etc.) for visual and functional comparisons.
    - **Supplier (1.1.1.2.2)**
      Brand-level segmentation (Toyota, BMW, Ford, etc.) to leverage manufacturer familiarity.
    - **Specifications (1.1.1.2.3)**
      Technical filters (color, fuel type, transmission, availability) to satisfy precision queries.

3. About Section (1.2)

A provenance layer that fosters user trust through transparent team and credit information.

- **Credit Details (1.2.1)**
  Attribution module detailing project leadership and contributors.
  - **About Us (1.2.1.1)**
    Profiles the team leader's expertise and role.
  - **Members (1.2.1.2–1.2.1.5)**
    Individual contributor bios underscoring cross-functional skills.

4. Product Detail View (1.3)

An information-rich page combining visual media and textual metadata to support purchase decisions.

- **Images (1.3.1)**
  High-resolution photo gallery showcasing vehicle exteriors and interiors.

- **Description (1.3.2)**
  Narrative summary highlighting unique selling points and feature overviews.

- **Category (1.3.3)**
  Contextual tags linking back to model, supplier, and specification facets.

Key Insights

- **Hierarchical Taxonomy**: Dual-axis classification (model type vs. supplier) empowers both exploratory and targeted search strategies.

- **Facet-Driven Discovery**: Specification filters function as precision levers, minimizing cognitive load in decision-making.

- **Provenance & Credibility**: Dedicated “About” metadata strengthens perceived system legitimacy.

- **Scalability**: Modular node design anticipates expansion to new vehicle classes, brands, or attribute dimensions.