Cégep de la Gaspésie
et des Îles

Automobile Store eCommerce Website

Final Project - CST 413
January 21, 2021

Under the guidance of
Ahmad Reza Madabadi
Sakkaravarthi Ramanathan

Submitted by
Simranjeet (1896064), Jasvir (1993324) and Gagandeep (1992589)

Table of Contents

1.1 Document Scope and Purpose

The purpose of the document is to serve as a guide to designers, developers, and testers who are responsible for the engineering of the **Automobile Store eCommerce Website** project. It should give engineers all of the information necessary to design, develop, and test the website.

The owner maintains as few inventories for each car as reasonable, to reduce inventory overheads after being inspired by "just in time (JIT) philosophy". When a customer places an order online, the order is placed to the company directly and the model is henceforth made available.

The system can be accessed by:

- Admin
- Customer

The customers and the admin have to authenticate before being able to access the site. Customers don't have the option for guest checkout or book test drive as guests.

1.2 General Idea

An automobile website sells the vehicles of several models and companies. The cars are available at the warehouse in one location and a team is taking care of the online orders. To streamline the sales and supply ordering, we are creating an eCommerce website. There is also an option for addons, modification of cars, etc.

1.3 Actors and their Roles

The various users that anticipate with this product are:

Admin:

- **Maintenance:** Maintaining the website and updating inventory or updating important information on the website.

- **View customer details:** View the personal details of the customer for orders.

- **Managing Sales to Customers:** Responsible for properly allocating the selected product according to the customer's choice and updating its status.

● **View Product Stocks:** Keep track of each product item's stocks for selling purposes.

Customer:

● **Login:** Customers must have a valid login ID to enter into the site.

● **Registration:** New users can sign up by creating a new ID.

● **Choosing and comparing products:** Can view all available products and can compare them and make a choice for purchasing products. A comparison is not made using any comparison page.

● **Purchasing and Test Drive:** Can purchase any product and make payment or book test drive.

● **Logout:** Customers can log out of the site.

1.4 Functional and Non-Functional Requirements

Functional Requirements

FR1	Customers can select each add-on of products like car color, tires, tire, and interior.
FR2	Customers can view the products that they intend to buy/purchase.
FR3	Customer can use their credit/debit card to make a payment
FR4	Customer can save select downpayment years and amount
FR5	Admin has access to the highest privileges on the system
FR6	Admin can view a log of all the customers' transactions and purchases
FR7	Admin can delete users which could be spam or ghost accounts
FR8	Admin can accept or reject and refund an order
FR9	Admin has access to shipping and billing details for verification
FR10	Admin can update the inventory manually
FR11	Admin can delete or put a product out of stock

FR12	Customer can book a test drive
FR13	Customer can check EMI for financing
FR14	Admin can accept or reject a Test Drive booking
FR15	Admin can add a new car companies and vehicles

The list for functional requirements is not exhaustive and more functional requirements can be added if the need arises.

Non Functional Requirements

- **Flexibility**

The system should be flexible enough to handle several additions and deletions of users/parts and support extensions in the functionalities if required.

- **Maintainability**

The website will be maintainable enough so that extra features can be added or modified at any time.

- **Testability**

The web site can be tested with several users using it at a time.

- **Reliability**

It will be reliable and will send the correct information to the correct user who has registered to the site. Since all users are familiar with the general usage of internet browsers, no special training is required.

- **Usability and portability**

The website will be easy to use and will be portable enough to transfer from one server machine to another. Being on the cloud, it will have a CDN that will serve it from the closest server to the user's location.

- **Availability**

Since the app is cloud-based, the system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

- **Efficiency Requirements**

Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less. Being cloud-based, it takes seconds to load everything up and be ready for the end-user.

- **Privacy**

Meet basic requirements for GDPR.

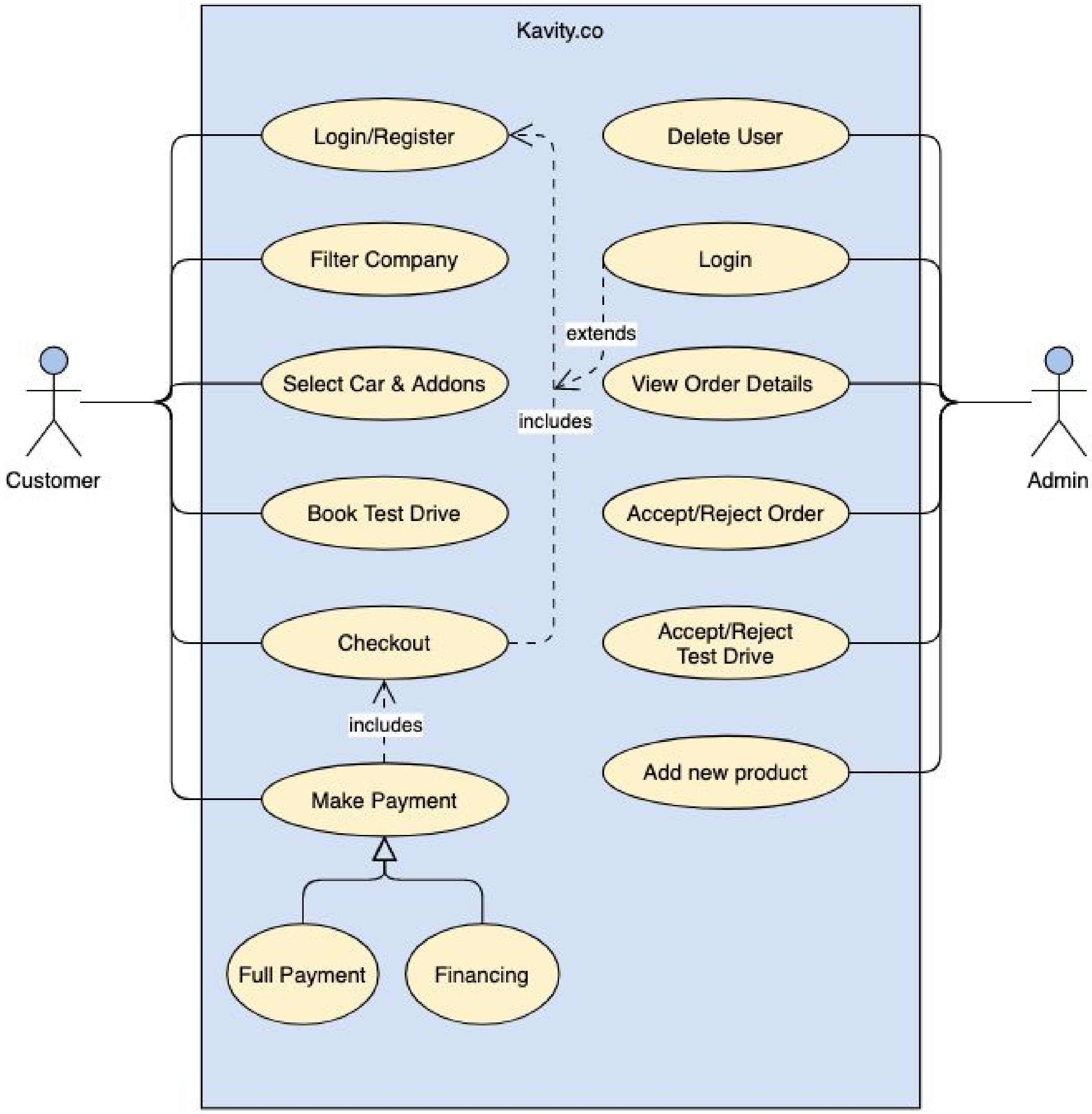
1.5 Login Details

The frontend for the website is located at https://kavity.co/
Customer username is sakku@123.com and password is sakku

The admin panel for the website is located at https://kavity.co/admin
Admin username is admin and password is 123456

The GitHub Repository is located at https://github.com/gagandeeply/automobile-store

1.6 Use Case Diagram



1.7 Use Case Scenarios

Use Case (UC01): Login

Primary Actor	Customer
Stakeholder & Interests	The customer can log in to the website
Pre-Condition	The customer’s device should be connected to the internet
Main Success	The customer should be successfully registered
Extension	If there is an error while logging in, the customer will be notified
Special Requirements	The text should be readable

Use Case (UC02): Filter Company

Primary Actor	Customer
Stakeholder & Interests	The Customer can filter products by company
Pre-Condition	The customer’s device should be connected to the internet
Main Success	Customer should successfully get the list of filtered products
Extension	If the customer cannot find the required product, they can remove the filter
Special Requirements	The text should be readable

Use Case (UC03): Select Car & Addons

Primary Actor	Customer
Stakeholder & Interests	The customer can choose the required product shown after filtering
Pre-Condition	The customer’s device should be connected to the internet
Main Success	The customer should be registered

Extension	If the customer cannot select the product, the product is out of stock
Special Requirements	The text should be readable

Use Case (UC04): Book Test Drive

Primary Actor	Customer
Stakeholder & Interests	The customer should be able to book a test drive
Pre-Condition	The customer must be registered, should have the internet on the device, and should have the proper amount to make a payment
Main Success	After the booking, there should be a confirmation pop-up “order confirmed”.
Extension	If the booking is not successful, it should mention ‘incomplete transaction’
Special Requirements	The text should be readable

Use Case (UC05): Checkout

Primary Actor	Customer
Stakeholder & Interests	The customer can purchase products from the website
Pre-Condition	The customer must be registered and should have the internet on the device
Main Success	Customer can see the success message after clicking
Extension	no-extensions
Special Requirements	The text should be readable

Use Case (UC06): Make a payment

Primary Actor	Customer
Stakeholder & Interests	The customer should be able to make a payment after checking out or finance/EMI the vehicle
Pre-Condition	The customer must be registered, should have the internet on the device, and should have the proper amount to make a payment

Main Success	After the payment, there should be a confirmation pop-up “order confirmed”.
Extension	If the payment is not successful, it should mention ‘incomplete transaction’
Special Requirements	The text should be readable

Use Case (UC07): Delete User

Primary Actor	Admin
Stakeholder & Interests	Admin
Pre-Condition	Admin must be logged in.
Main Success	Admin is able to delete customer account
Extension	No extension
Special Requirements	The text should be readable

Use Case (UC08): Login

Primary Actor	Admin
Stakeholder & Interests	The Admin can log in to the website
Pre-Condition	The Admin’s device should be connected to the internet
Main Success	The Admin should be successfully registered
Extension	If there is an error while logging in, the Admin will be notified
Special Requirements	The text should be readable

Use Case (UC09): View Order Details

Primary Actor	Admin
Stakeholder & Interests	Admin can view log of the customer’s order
Pre-Condition	The Admin’s device should be connected to the internet
Main Success	The Admin can view the logs generated by the system
Extension	No extension
Special Requirements	The text should be readable

Use Case (UC10): Accept/Reject Order

Primary Actor	Admin
Stakeholder & Interests	Admin can manage the Customer order
Pre-Condition	The Admin’s device should be connected to the internet
Main Success	The Admin can either accept or reject a customer order based on the stock level and company requirements
Extension	Admin can keep the orders maintained
Special Requirements	The text should be readable

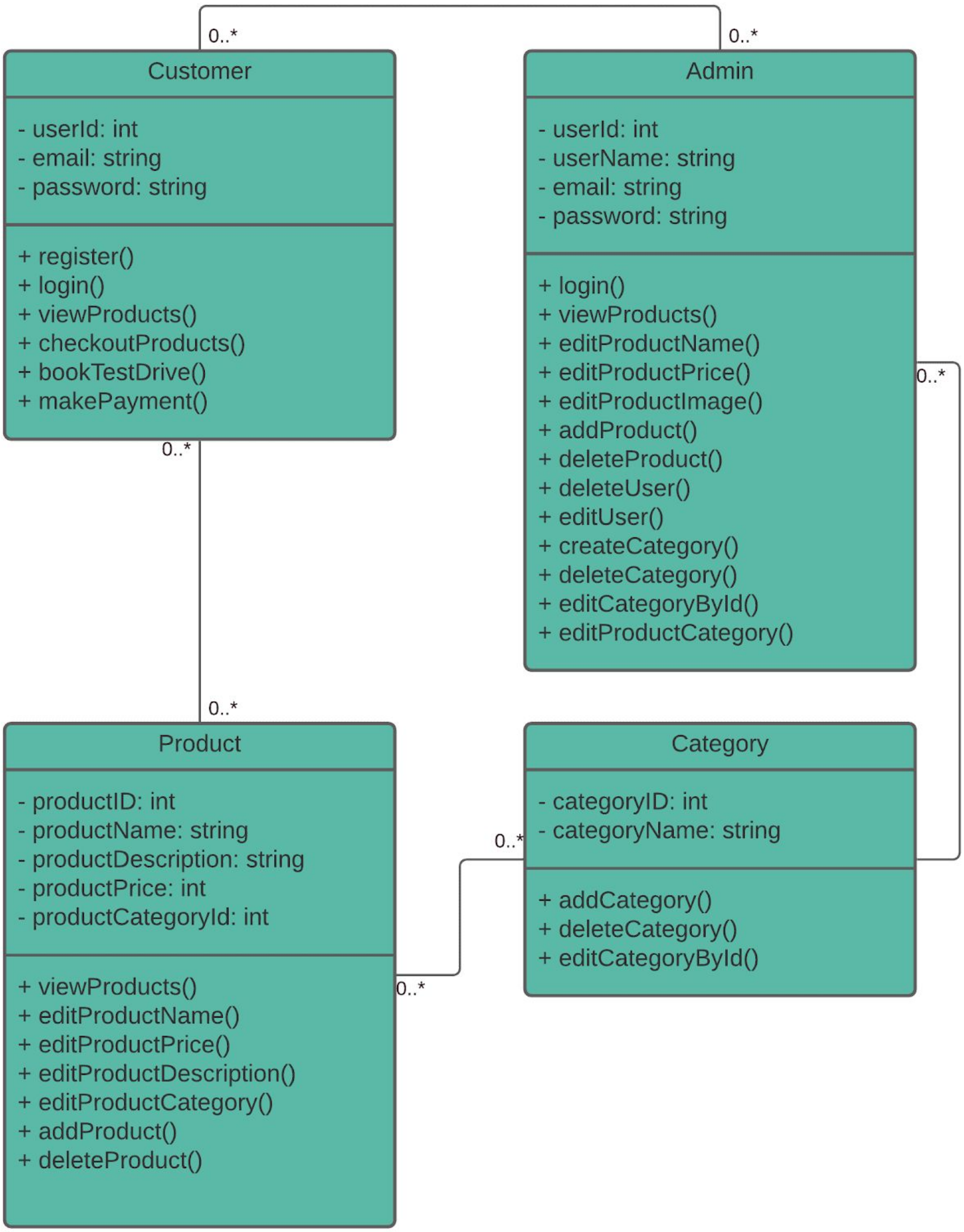
Use Case (UC10): Accept/Reject Test Drive

Primary Actor	Admin
Stakeholder & Interests	Admin can manage the test drive booking
Pre-Condition	The Admin’s device should be connected to the internet
Main Success	The Admin can either accept or reject a test drive
Extension	Admin can keep the test drive record maintained
Special Requirements	The text should be readable

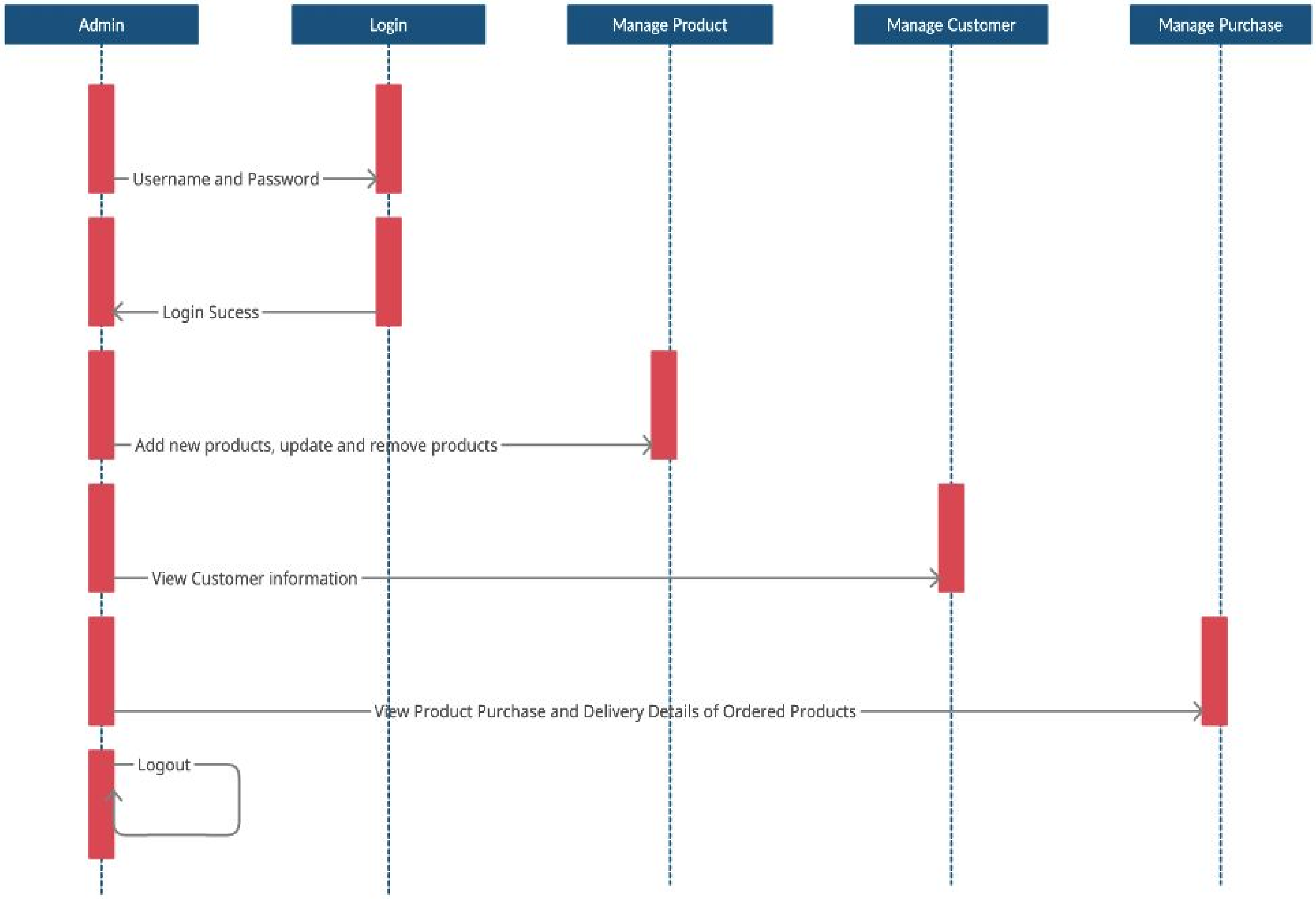
Use Case (UC11): Add New Product

Primary Actor	Admin
Stakeholder & Interests	Admin, Customer
Pre-Condition	The Admin’s device should be connected to the internet
Main Success	The Admin can generate bill of payment for the customer
Extension	Admin can add a new product in the stock
Special Requirements	The text should be readable

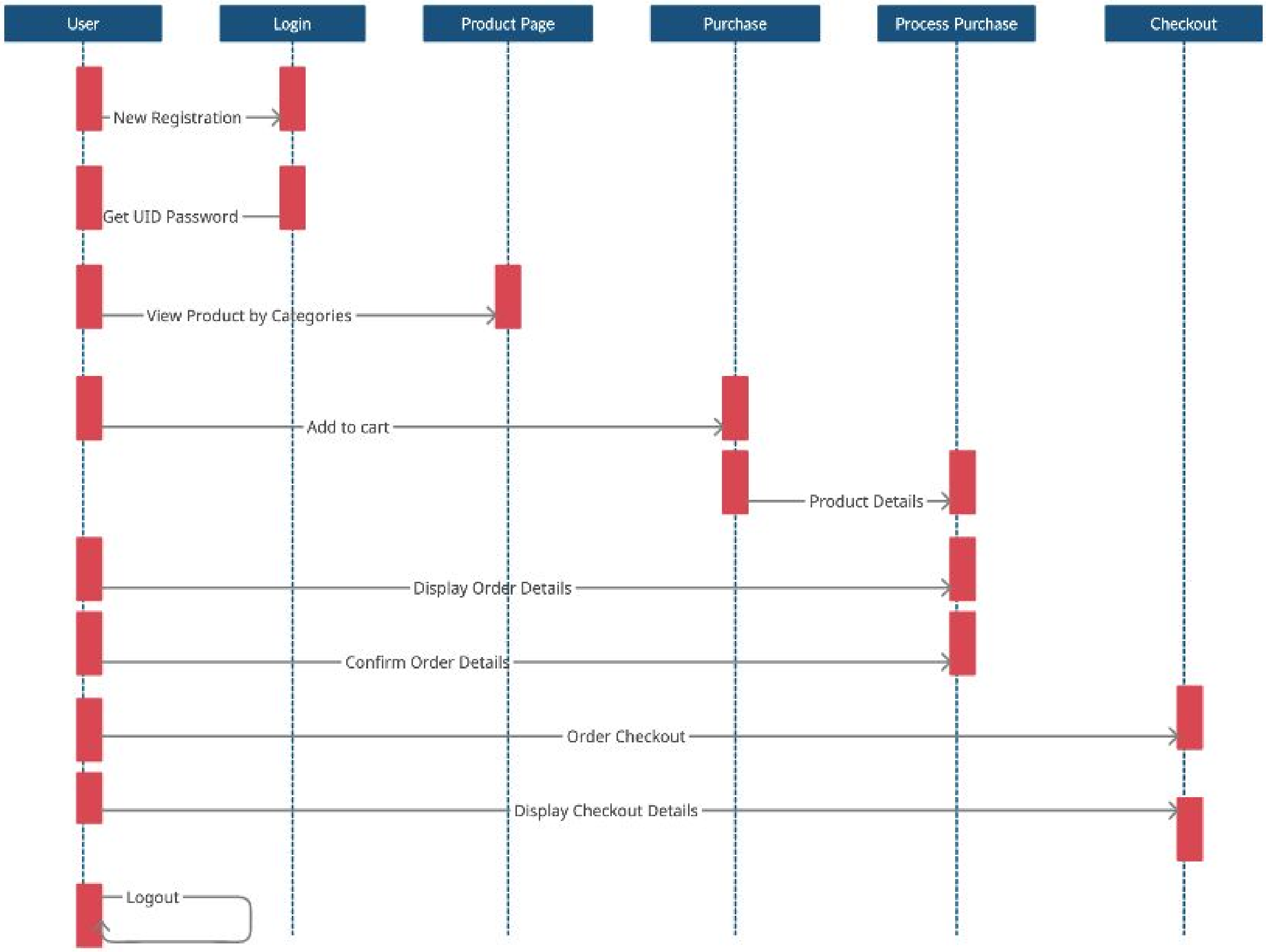
1.8 Class Diagram



1.9 Sequence Diagram


Admin
Login
Manage Product
Manage Customer
Manage Purchase
Username and Password
Login Sucess
Add new products, update and remove products
View Customer information
View Product Purchase and Delivery Details of Ordered Products
Logout

SD01: Sequence diagram for admin



SD02: Sequence diagram for customer

1.10 Database Details

The database consists of 6 tables: admin, category, product, users, test_book, order_table. Each of these tables is described in detail below:

Table: admin

The ‘admin’ tables consists of 9 columns (with data types as):

- id: int(11)
- username: varchar(225)
- email: varchar(225)
- password: varchar(225)
- status: varchar(225)
- role: varchar(225)
- date: datetime(6)
- last_login: varchar(225)
- last_logout: varchar(225)

‘id’ is the primary key of the table ‘admin’.

Table: category

The ‘category’ table consists of 4 columns (with data types as):

- id: int(11)
- cname: varchar(225)
- status: varchar(225)
- date: timestamp

‘id’ is the primary key of the table ‘category’. ‘date’ column has a datatype of timestamp to render the user’s location time as it converts the current time to UTC depending on the user's location.

Table: product

The ‘product’ table contains 8 columns (with data types as):

- id: int(11)
- pname: varchar(500)
- pdescription: varchar(2000)
- pprice: varchar(225)
- pcategory: varchar(225)
- pimage: varchar(500)
- date: timestamp
- status: varchar(11)

The primary key in the table is ‘id’.

Table: users

The ‘product’ table contains 7 columns (with data types as):

- id: int(11)
- name: varchar(225)
- email: varchar(225)
- phone: varchar(225)
- password: varchar(225)
- status: varchar(225)
- date: timestamp

The primary key in the table is ‘id’.

Table: test_book

The test_book table contains 9 columns (with data types as):

- id: int(11)
- name: varchar(225)
- p_id: varchar(225)
- phone: varchar(225)
- address: varchar(225)
- bookdate: varchar(225)
- user_id: varchar(225)
- status: varchar(225)
- date: timestamp

The primary key in the table is ‘id’.

Table: order_table

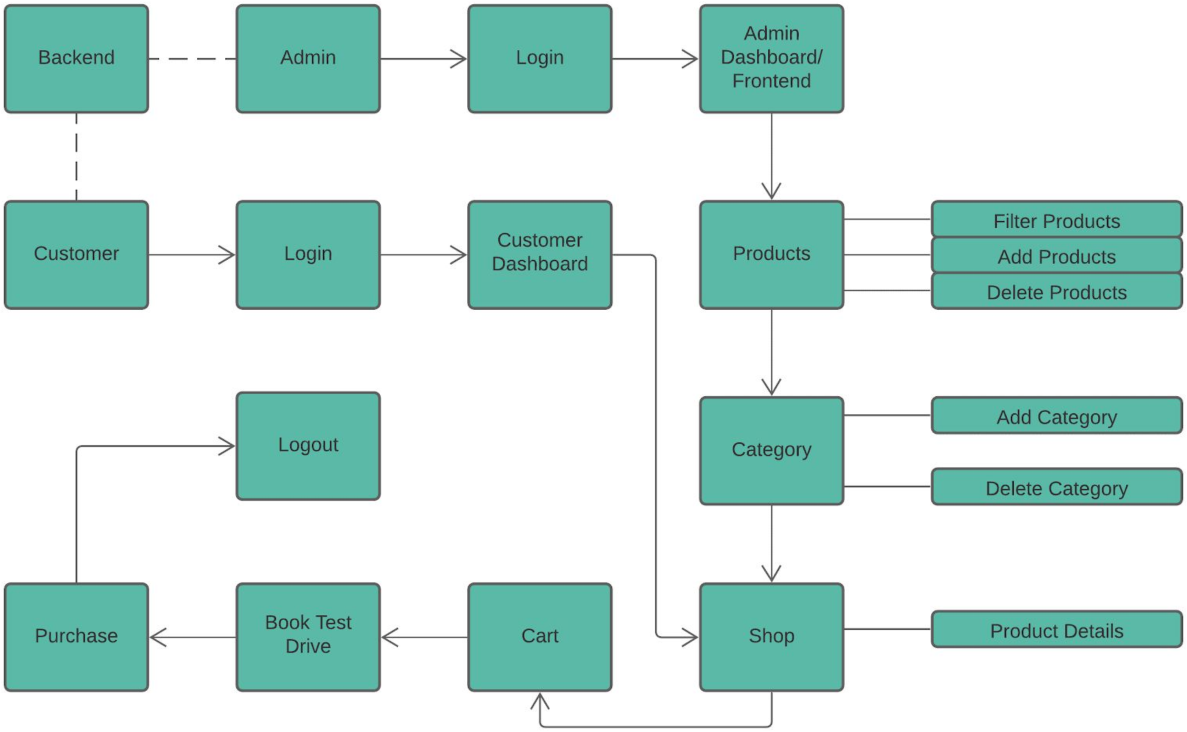
The order_table table contains 22 columns (with data types as):

- id: int(11)
- user_id: varchar(225)
- pid: varchar(225)
- pname: varchar(225)
- proprice: varchar(225)
- color: varchar(225)
- tire: varchar(225)
- interior: varchar(225)
- firstName: varchar(225)
- lastName: varchar(225)
- phone: varchar(225)
- email: varchar(225)
- address: varchar(225)
- address2: varchar(225)

- country: varchar(225)
- state: varchar(225)
- zip: varchar(225)
- quantity_input: varchar(225)
- status: varchar(225)
- date: timestamp
- downp: varchar(225)
- downmonth: varchar(255)

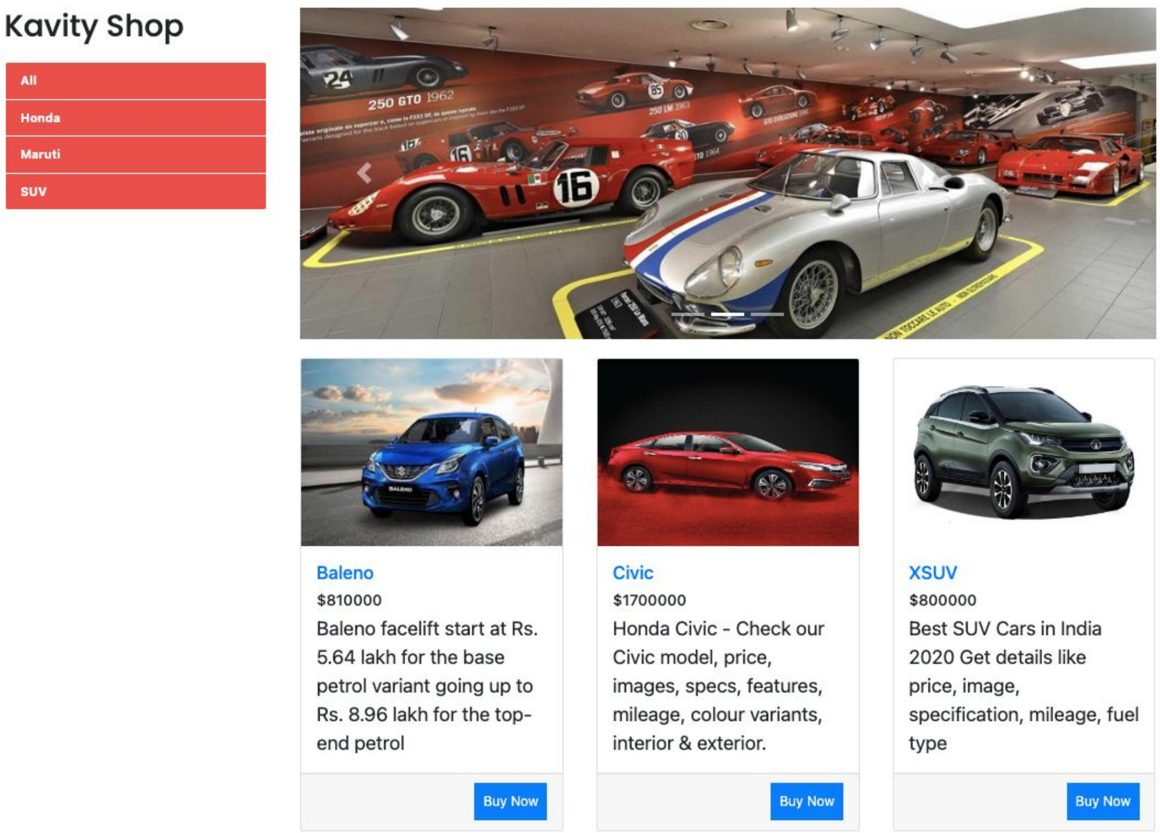
The primary key in the table is ‘id’.

1.11 Screen Flow Diagram



2. User Interface Screens

- 2.1 Shop Page



Kavity Shop

Baleno

$810000

Baleno facelift start at Rs. 5.64 lakh for the base petrol variant going up to Rs. 8.96 lakh for the top-end petrol

Civic

$1700000

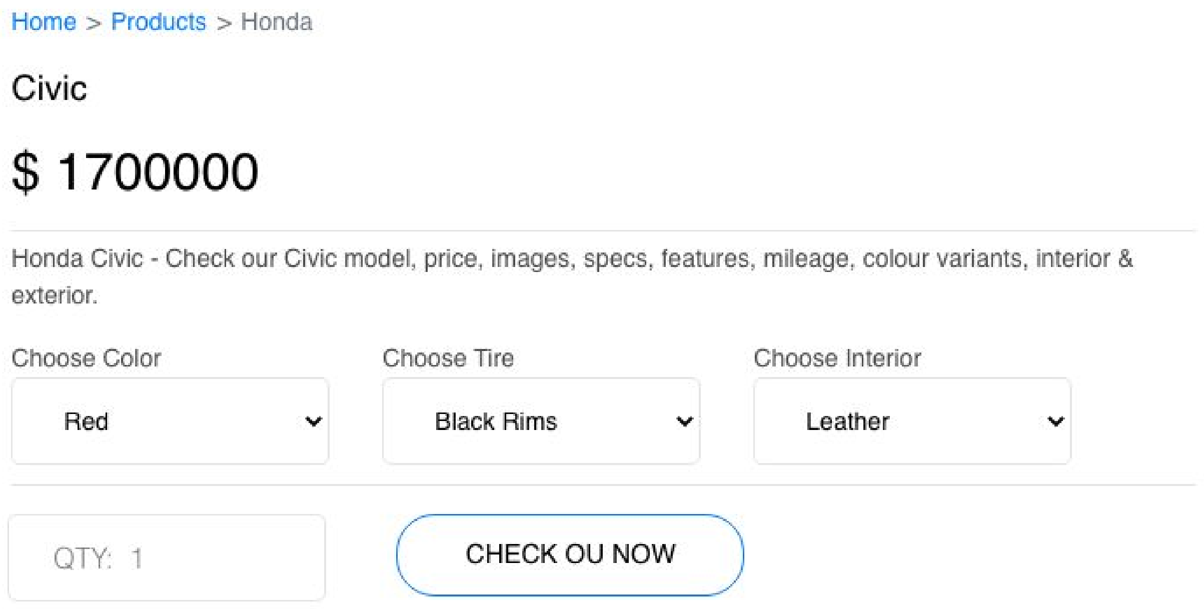
Honda Civic - Check our Civic model, price, images, specs, features, mileage, colour variants, interior & exterior.

XSUV

$800000

Best SUV Cars in India 2020 Get details like price, image, specification, mileage, fuel type

- 2.2 Product Page



Home > Products > Honda

Civic

$ 1700000

Honda Civic - Check our Civic model, price, images, specs, features, mileage, colour variants, interior & exterior.

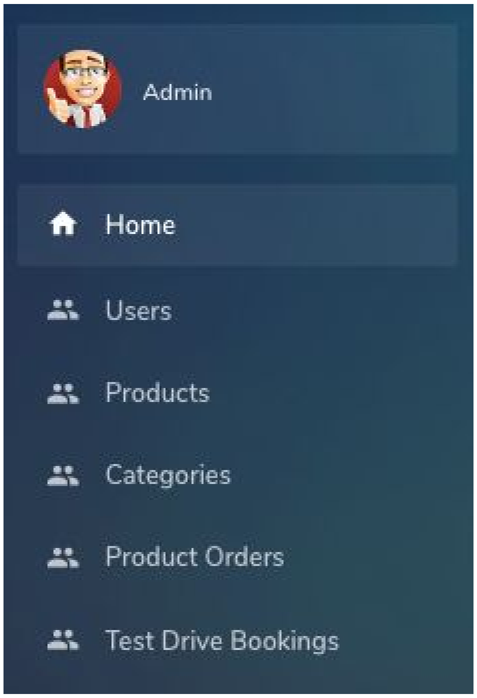
Choose Color: Red

Choose Tire: Black Rims

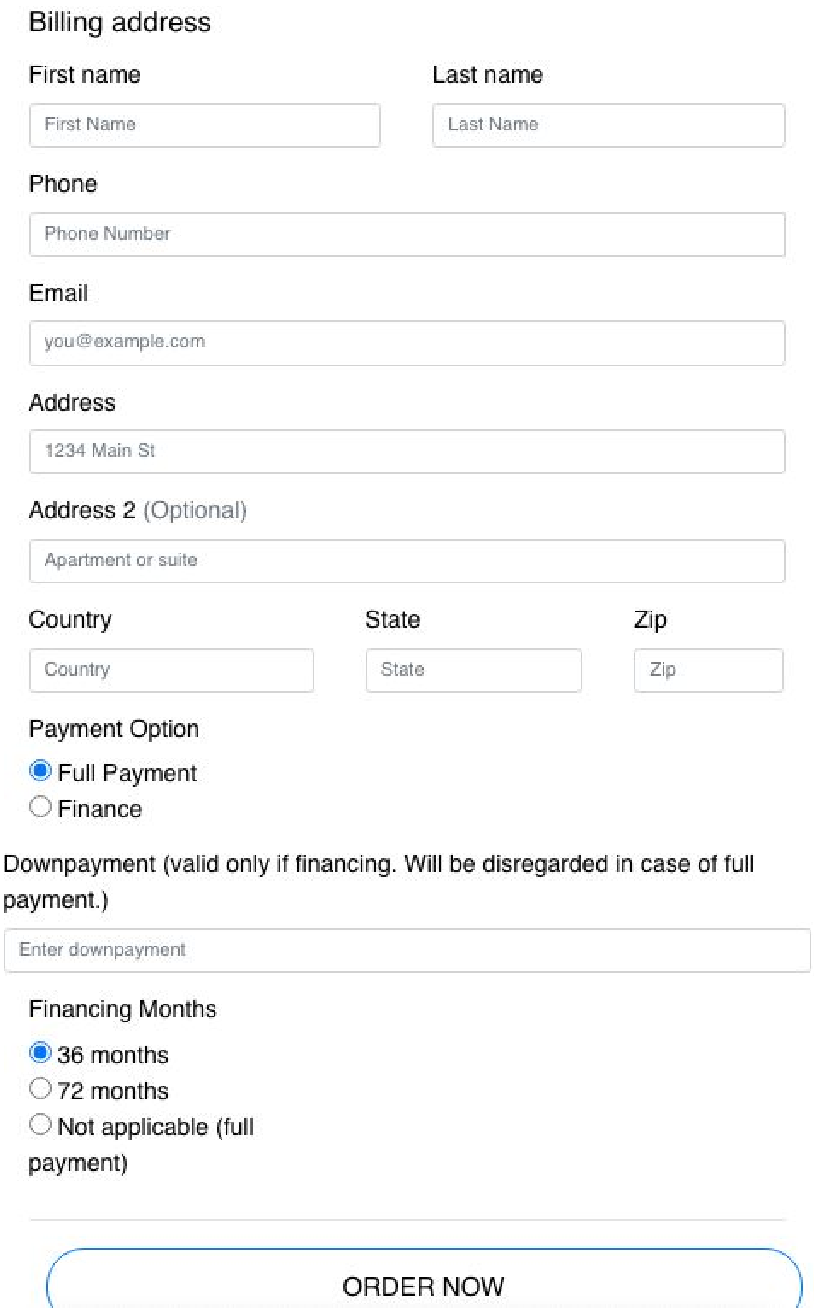
Choose Interior: Leather

QTY: 1

- 2.3 Admin Panel



- 2.4 Checkout Page



Billing address

First name: First Name

Last name: Last Name

Phone: Phone Number

Email: you@example.com

Address: 1234 Main St

Address 2 (Optional): Apartment or suite

Country: Country

State: State

Zip: Zip

Payment Option

● Full Payment
○ Finance

Downpayment (valid only if financing. Will be disregarded in case of full payment.)

Enter downpayment

Financing Months

● 36 months
○ 72 months
○ Not applicable (full payment)

3. Unit Testing

3.1 Function: Admin Login

ID	Test Case	Input Value	Output	Result
1	Logging in as admin			
1.1	When the User ID and password field is empty	Login with credentials	Alert user to “All Field Required!	Pass
1.2	When an invalid ID and password is entered	Login with credentials	Alert user to “All Field Required!	Pass
1.3	When User ID and password are not matching	Login with credentials	Alert user to “All Field Required!	Pass
1.4	When User ID and password are matching	Login with credentials	Alert user to “All Field Required!	Pass

3.2 Function: Create a category

ID	Test Case	Input Value	Output	Result
2	Creating a category in the backend			
2.1	Creating a category with	Login with	Category	Pass

	status 1	credentials	creation successful	
2.2	Creating a category with status, not 1	Login with credentials	Admin panel not accessible	Pass

3.3 Function: Delete Category

ID	Test Case	Input Value	Output	Result
3	Deleting category in backend			
3.1	Deleting category with status 1	Login with credentials	Category deletion successful	Pass
3.2	Deleting category with status, not 1	Login with credentials	Admin panel not accessible	Pass

3.4 Function: Create Product

ID	Test Case	Input Value	Output	Result
2	Creating a product in the backend			
2.1	Creating a product with status all data	Enter all data	product creation successful	Pass
2.2	Creating a product with some fields empty	Enter some data	Please fill out this field	Pass

3.5 Function: Delete Product

ID	Test Case	Input Value	Output	Result
4	Deleting product in backend			
4.1	Deleting product with status 1	Login with credentials	Product deletion successful	Pass
4.2	Deleting product with status, not 1	Login with credentials	Admin panel not accessible	Pass

3.6 Function: Register

ID	Test Case	Input Value	Output	Result
5	Register from frontend			
5.1	No value in the name text field	Name empty	Request unsuccessful, output: All Field Required!	Pass
5.2	No value in the email text field	Email empty	Request unsuccessful, output: All Field Required!	Pass
5.3	No value in the phone text field	Phone empty	Request unsuccessful,	Pass

			output: All Field Required!	
5.4	No value in the password text field	Password empty	Request unsuccessful, output: All Field Required!	Pass
5.5	All values filled	All values filled	Registration successful	Pass

3.7 Function: User Login

ID	Test Case	Input Value	Output	Result
1	Customer Login			
1.1	Login with username empty and password added.	Username: Pass: 123456	Login error	Pass
1.2	Login with username using whitespace and password added.	Username: admin Pass: 123456	Login error	Pass
1.3	Login with correct username and password convention	Username: admin Pass: 123456	Login successful	Pass

3.8 Function: Checkout

ID	Test Case	Input Value	Output	Result
2	Checkout from frontend			

2.1	Fill all details in the checkout form	Enter all data	Checkout form submitted	Pass
2.2	Some fields missing in the checkout form	Some fields missed	All Fields Required!	Pass

3.9 Function: Book Test Drive

ID	Test Case	Input Value	Output	Result
2	Test Drive form from frontend			
2.1	Fill all details in the Test Drive form	Enter all data	Checkout form submitted	Pass
2.2	Some fields missing in the Test Drive form	Some fields missed	All Fields Required!	Pass

4. Integration Testing

4.1 Add to cart

ID	Test Name	Modules Tested	Input Value	Expected Outcome	Result
1.1	Add to cart	Login and Payment	User logged in while buying	Payment Successful	Pass
1.2	Add to cart	Login and Payment	User not logged in while buying	Payment unsuccesful	Pass

4.2 Book a test drive

ID	Test Name	Modules Tested	Input Value	Expected Outcome	Result
1.1	Test drive with login	Login and submit the form	User logged in while test drive submission	Booking Successful	Pass
1.2	Test drive without login	Login and submit the form	User not logged in while submitting the form	Booking unsuccesful	Pass

5. RESTful Methods

5.1 Use Case Name: Register

<table>
<tr><td>Use Case ID</td><td>1</td></tr>
<tr><td>Use Case Name</td><td>Register</td></tr>
<tr><td>URL</td><td>/register.php</td></tr>
<tr><td>Restful Method</td><td>POST</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Action</td><td>Type</td><td>Description</td></tr>
<tr><td>fname</td><td>string</td><td>Name</td></tr>
<tr><td>email</td><td>string</td><td>email</td></tr>
<tr><td>phone</td><td>int</td><td>Phone number</td></tr>
<tr><td>passw</td><td>string</td><td>password</td></tr>
</table>
</td></tr>
<tr><td>Response</td><td>**Success -**
success: function (status) {
var status=$.trim(status);</td></tr>
</table>

	if(status=='2')
	{
	$("#error").html('Email Already Exist!');
	}
	else if(status=='1')
	{
	$("#error").html('Register Successfully Welcome');
	document.location.href = "https://kavity.co"
	}
	Failure -
	else
	{
	$("#error").html('Invalid credentials !');
	}

5.2 Use Case Name: Login

<table>
<tr><td>Use Case ID</td><td>2</td></tr>
<tr><td>Use Case Name</td><td>Login</td></tr>
<tr><td>URL</td><td>/login.php</td></tr>
<tr><td>Restful Method</td><td>POST</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Action</td><td>Type</td><td>Description</td></tr>
<tr><td>uname</td><td>string</td><td>Username</td></tr>
<tr><td>pass</td><td>string</td><td>password</td></tr>
</table>
</td></tr>
<tr><td>Response</td><td><b>Success</b> -
success: function (status) {</td></tr>
</table>

	var status=$.trim(status);
	if(status=='1')
	{
	document.location.href = "https://kavity.co"
	}
	Failure -
	else
	{
	$("#error").html('Invalid credentials !')
	}

5.3 Use Case Name: Create Order

<table>
<tr><td>Use Case ID</td><td>3</td></tr>
<tr><td>Use Case Name</td><td>Create Order</td></tr>
<tr><td>URL</td><td>/process.php</td></tr>
<tr><td>Restful Method</td><td>POST</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Action</th><th>Type</th><th>Description</th></tr>
<tr><td>firstname</td><td>string</td><td>Firstname</td></tr>
<tr><td>lastname</td><td>string</td><td>Last name</td></tr>
<tr><td>phone</td><td>int</td><td>Phone number</td></tr>
<tr><td>email</td><td>string</td><td>Email address</td></tr>
</table>
</td></tr>
</table>

address	string	Physical Address
country	string	Country
state	string	State/Province
zip	string	Zipcode

Response

Success -

```
function (status) {

                                    var status=$.trim(status);
       if(status=='1')

                           {
           $("#error").html('Your Order Placed Successfully');
                                   document.location.href                          =
"https://kavity.co/shop.php"

                           }
```

Failure -

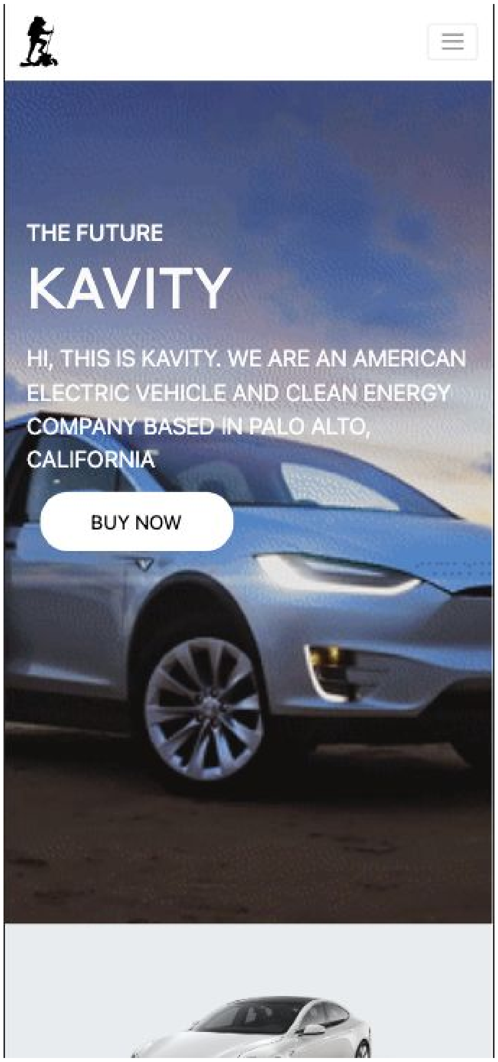
```
         else

                           {

                                   $("#error").html('Invalid credentials !');

                           }

         else

         {

                  $("#error").html('All Field Required !')

         }
```

5.4 Use Case Name: Book Test Drive

<table>
<tr><td>Use Case ID</td><td>4</td></tr>
<tr><td>Use Case Name</td><td>Book Test Drive</td></tr>
<tr><td>URL</td><td>/process.php</td></tr>
<tr><td>Restful Method</td><td>POST</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Action</td><td>Type</td><td>Description</td></tr>
<tr><td>name</td><td>string</td><td></td></tr>
<tr><td>number</td><td>string</td><td></td></tr>
<tr><td>address</td><td>string</td><td></td></tr>
<tr><td>bdate</td><td>int</td><td></td></tr>
</table>
</td></tr>
<tr><td>Response</td><td><b>Success -</b>
function (status) {
var status=$.trim(status);
//$("#errorsu"+x).html('Your Booking Submit Successfully!')
// $("#name"+x).val('');
// $("#pnumber"+x).val('');
// $("#address"+x).val('');
// $("#bdate"+x).val('');
document.location.href = "https://kavity.co/shop.php"
}
<b>Failure -</b>
else</td></tr>
</table>

	{
	$("#error"+x).html('All Field Required !')
	}

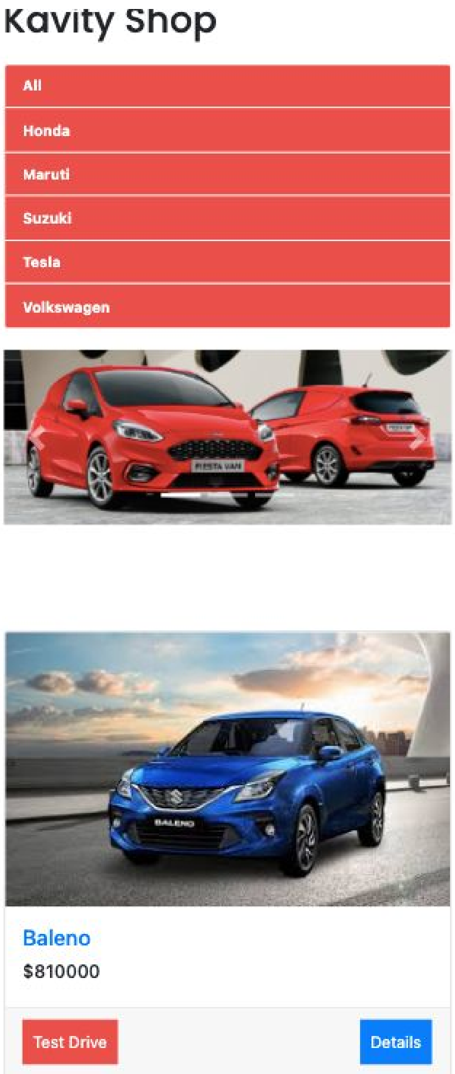
6. Visual Testing



THE FUTURE

KAVITY

HI, THIS IS KAVITY. WE ARE AN AMERICAN ELECTRIC VEHICLE AND CLEAN ENERGY COMPANY BASED IN PALO ALTO, CALIFORNIA



Kavity Shop

Baleno

$810000

The site is completely responsive on the front-end. It has been tested on multiple screens and an example of iPhone X resolution is above two images, where the site adapts to the screen responsively.

7. Test Coverage

Work Pack	Kavity eCommerce Website
Tester	Simranjeet, Jasvir
Date	3 Jan 2021
Date Tested	11 Jan 2021
Total Test Cases	28
No. of Cases Passed	28
No. of Cases Failed	0
Bugs Found	7
Bugs Awaiting Fix	0
Bugs Fixed	7
% coverage	N/a

8. Responsibilities

The project was completed in multiple stages and sprints with a lot of trial and error and would not have been successful had it not been a team effort. The responsibilities of different team members are as follows:

- Jasvir Kaur
    - Designing pages - Homepage, Shop, Product Page, Login & Register Page and admin dashboard
    - Visual Testing
    - Integration Testing and Test Coverage
    - Screen Flow Diagrams and final presentation
- Simranjeet Kaur
    - Designing pages - Homepage, Shop, Product Page, Login & Register Page and admin dashboard
    - Making them responsive
    - Preparing the final presentation
    - Integration Testing and Test Coverage
- Gagandeep Singh
    - Coding the functionality in the pages and documentation

- Unit Testing
- Regression Testing
- Setting up Hosting, Server, Domain, FTP and Database and tables

9. Conclusion

The project entitled Kavity is effectively completed through its design, development, and implementation, and testing. It successfully fulfilled the objective of efficient web development.

The application provides a shopping portal for the customer and admin by saving their valuable time and implementation effort. The admin has full access to the system. He can manage products, customer information, and payment information. The user can shop for cars and addons by viewing the products online, adding them to cart, and efficiently making purchases. The customer has the option to calculate the EMI in case they want to finance the vehicle and has the option to select either while checking out. They can also book test drives which the admin can approve or deny.

Working on the project was a challenging task and it was one of the finest learning opportunities we got. It has opened us up to a lot of new concepts in development and testing and also their implementation in the real world.