INTRODUCTION

1. Introduction

General Overview

The project "*Resell Hub"* is a web based ecommerce tool which is important to an online marketplace where people can buy and sell goods and services. The software allows users to post advertisements for products or services they want to sell, and buyers can browse through these listings to find items they are interested in purchasing. The application provides a platform for users to communicate with each other, negotiate prices, and arrange for payment and delivery .It is a C2C type of commercial website.

It is designed to be user-friendly and accessible, with a simple interface that makes it easy for anyone to use. The application is also secure, with measures in place to protect users from fraud and other types of online scams. Overall, this website provides a convenient and efficient way for people to buy and sell goods and services online. To connect buyers and sellers from around the world, offering a convenient and secure way to conduct transactions.To create a community where users can find unique and hard-to-find items, as well as new and used products.To provide a platform for entrepreneurs and small businesses to reach a global audience.To facilitate e-commerce and empower individuals and businesses to succeed in the online marketplace.

This whole project is designed in Laravel framework and different variables and strings have been used for the development of this project. It's easy to operate and understand by the users. Here, Customers can create their account and place order directly. And by the help of this system home delivery of the products can also be possible with the help of orders database. This project consists of proper Validations for the admin and users with proper login system. It is easy to upload, view, search and delete item details on the website.

The website that is similar to E-bay and Hamrobazar.com, an online marketplace where people can buy and sell their old and used secondhand goods and services. The website allows users to post advertisements for products or services they want to sell, and buyers can browse through these listings to find items they are interested in purchasing .This website contains a register page where buyer and seller must registered with email and also the seller can post and advertise their products here according to their choice and price .The buyer can contact from seller with the help of contact number to buy product they like to buy. And with other various features.

1.1 Objective:

- ❖ To connect buyers and sellers from around the world, offering a convenient and secure way to conduct transactions.
- ❖ To create a community where users can find unique and hard-to-find items, as well as new and used products.
- ❖ To provide a platform for entrepreneurs and small businesses to reach a global audience.
- ❖ To facilitate e-commerce and empower individuals and businesses to succeed in the online marketplace.

1.2 Functionalities of the system:

- User Accounts: Resell Hub allows users to create accounts, which they can use to post listings, track their transactions, and communicate with other users.

- Search and Filter Tools: The system includes search and filter tools that allow users to easily find the products or services they are looking for, based on criteria such as location, price range, and category.

- Listing Creation: Sellers can create listings for their products or services, including images, descriptions, and price information.

- Messaging System: This system includes a messaging system that allows users to communicate with each other, negotiate prices, and arrange for the exchange of goods or services.

- Rating and Review System: Users can leave ratings and reviews for sellers and buyers, helping to build trust within the system community.

- Safety Tips and Guidelines: The website provides safety tips and guidelines for users to follow when buying or selling items, to help prevent fraud and ensure a safe and secure transaction.

- Mobile App: Resell Hub offers a mobile app that allows users to access the platform from their smartphones or tablets, making it convenient to buy and sell on the go.

- Payment Options: This system may offer various payment options, such as cash on delivery or online payment methods, to facilitate transactions between buyers and sellers.

- Notifications: Users can receive notifications for new messages, offers, and listings, keeping them updated on their activity on the platform.

- Favorites List: Users can save their favorite listings to a favorites list for easy access and comparison.

- Location-Based Services: Resell Hub uses location-based services to show users listings that are near their current location, making it easier to find local products and services.
- Ad Promotion: Sellers have the option to promote their listings for a fee, increasing the visibility of their products or services to potential buyers.

- Customer Support: Resell Hub provides customer support to assist users with any issues or questions they may have while using the platform.

1.3 Advantages

- **Wide Reach and Visibility:** These platforms have a large user base and wide reach, allowing your advertisements to be seen by a vast audience of potential buyers or customers.

- **User-Friendly Interface:** The platforms are designed to be user-friendly, making it easy for individuals and businesses to post advertisements without much technical expertise.

- **Free or Affordable Listings:** Resell hub, Quikr , and Just Dial typically offer free or cost-effective listing options, allowing sellers to advertise their products or services without significant upfront costs.

- **Diverse Categories:** The platforms cover a broad range of categories, from electronics and vehicles to real estate and services, providing options for various types of advertisements.

- **Location-Based Targeting:** These platforms often allow users to target specific geographic areas, ensuring that the advertisements reach the relevant local audience.

- **Quick Response:** Listings on these platforms can attract inquiries and responses from potential buyers or customers relatively quickly, making them suitable for time-sensitive sales or service offerings.

1.4 Disadvantages

- **Lack of privacy:** Lack of privacy is a serious disadvantage of e-commerce. A customer has to provide his personal details before making a purchase like address, name, and phone number and so on.

- **Lack of Personal Touch:** Some customers appreciate the personal touch they offer when visiting a physical store by interacting with the sales associates. Such personal touch is especially essential for businesses that sell high-end products as customers will want to buy the products and have an excellent experience during the process.
- **Product and Price Comparison:** With online shopping, customers can compare several products and find the least price. This forces many businesses to compete on price and reduce their profit margin, reducing the quality of products.

- **Need for Access to the Internet:** This is obvious, but don't forget that the customers do need access to the Internet before purchasing from any business! As many e- commerce platforms have the features and functionalities which require a high-speed Internet connection for an optimal consumer experience, there's a chance that companies are excluding visitors who have slow internet connections.

- **Credit Card Fraud:** Credit card frauds are a natural and growing problem for online businesses. It can lead to many chargebacks, which result in the loss of penalties, revenue, and a bad reputation.

- **IT Security Issues:** More and more organizations and businesses have fallen prey to malicious hackers who have stolen information of the customers from their databases. This could have financial and legal implications, but it also reduces the company's trust.

1.5 Scopes:

This is an online marketplace that facilitates the buying and selling of goods and services .It is an online marketplace that facilitates the buying and selling of goods and services. often provides a platform for local community discussions and events. It includes a section for real estate, enabling users to list and find properties for sale or rent. Some versions of Resell Hub have a job portal where users can find job listings or post job vacancies. Users can offer and find various services such as home repair, tutoring, event planning, and more. It typically has mobile apps for easy access and listing from smartphones.

2. SYSTEM ANALYSIS

Here are some key aspects of system analysis:

- ❖ Requirements Gathering: Identify and document the functional and non-functional requirements of the Resell- Hub Application.
- ❖ Use Case Analysis: Define the various use cases and user scenarios the application should support. Identify primary actors (users[Buyer and Seller], administrators).
- ❖ Data Modeling: Design the database schema to support the storage and retrieval of relevant data.
- ❖ Non-functional Requirements: Outline security measures, including user authentication, data encryption, and secure payment processing.
- ❖ Security Analysis: Evaluate potential security threats and establish measures to mitigate risks. Conduct a security assessment, including penetration testing.

a. System Requirements:

A. Hardware Requirements:

- Processor: Dual-core processor (2.5 GHz or higher)
- RAM: 8GB DDR4 or higher
- Storage: 128GB SSD or higher
- Display: 1920x1080 resolution or higher

B. Software Requirements:

- Operating System: Windows 10 Pro, 11….
- Web Server: XAMPP 7.4.3 or later
- Web Browser: Google Chrome
- Code Editor: Visual Studio Code 1.60.2

b. IMPLEMENTING TOOLS:

I. PHP:

PHP is an **HTML**-embedded, server-side scripting language designed for web development. It is also used as a general purpose programming language. It was created by **Rasmus Lerdorf** in 1994 and appeared in the market in 1995. Much of its syntax is borrowed from C, C++, and **Java. PHP** codes are simply mixed with HTML codes and can be used in combination with various web frameworks. Its scripts are executed on the server. PHP code is processed by a PHP interpreter. The main goal of PHP is to allow web developer to create dynamically generated pages quickly. A PHP file consists of texts, **HTML** tags and scripts with a file extension of .php, .php3, or .phtml. You can create a login page, design a form, create forums, dynamic and static websites and many more with PHP.

- **Why we use PHP?**

  You have obviously head of a number of programming language out there you may be wondering why we should want to use PHP as our poison for the web programming. Below are some of the compelling reasons.

  1. PHP is open source and free.
  2. Large community document.
  3. It is regular updated to keep abreast with the latest technology trends.

II. HTML

HTML stands for **HyperText Markup Language**. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes.

. **HTML Example:**

```
<!DOCTYPE>
<html>
<head>
<title>Web page title</title>
</head>
<body>
<h1>Write Your First Heading</h1>
<p>Write Your First Paragraph.</p>
</body>
</html>
```

- Why HTML is used in web pages:

Web developing includes two main section.

Backend: codes that are written **by Python, PHP, ASP.Net,** and **Go** language to name but a few by the developer.

Frontend: which is makeup showed by clients or users browsers and for doing this we should use HTML (Hypertext Markup Language), it just shows some elements for users and doesn't run any functions.

III. CSS:

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation and layout of HTML documents. It defines how elements should be displayed on the screen, in print, or in other media. CSS enables web developers to control the appearance of web pages, ensuring consistency and flexibility in the design across different devices and screen sizes.

IV. JavaScript:

JavaScript is a high-level, versatile programming language primarily known for its use in web development. It allows developers to add interactivity, manipulate the DOM (Document Object Model), handle events, and create dynamic content on websites. JavaScript is an essential component of modern web development and is supported by all major web browsers.

V. Security:

Security in the context of web development involves implementing measures to protect websites, web applications, and users from various threats and vulnerabilities. It's a critical aspect of the development process to ensure the confidentiality, integrity, and availability of data. HTTPS is implemented for secure communication, and security practices include input validation, parameterized queries, and protection against SQL injection and cross-site scripting (XSS).

VI. Laravel:

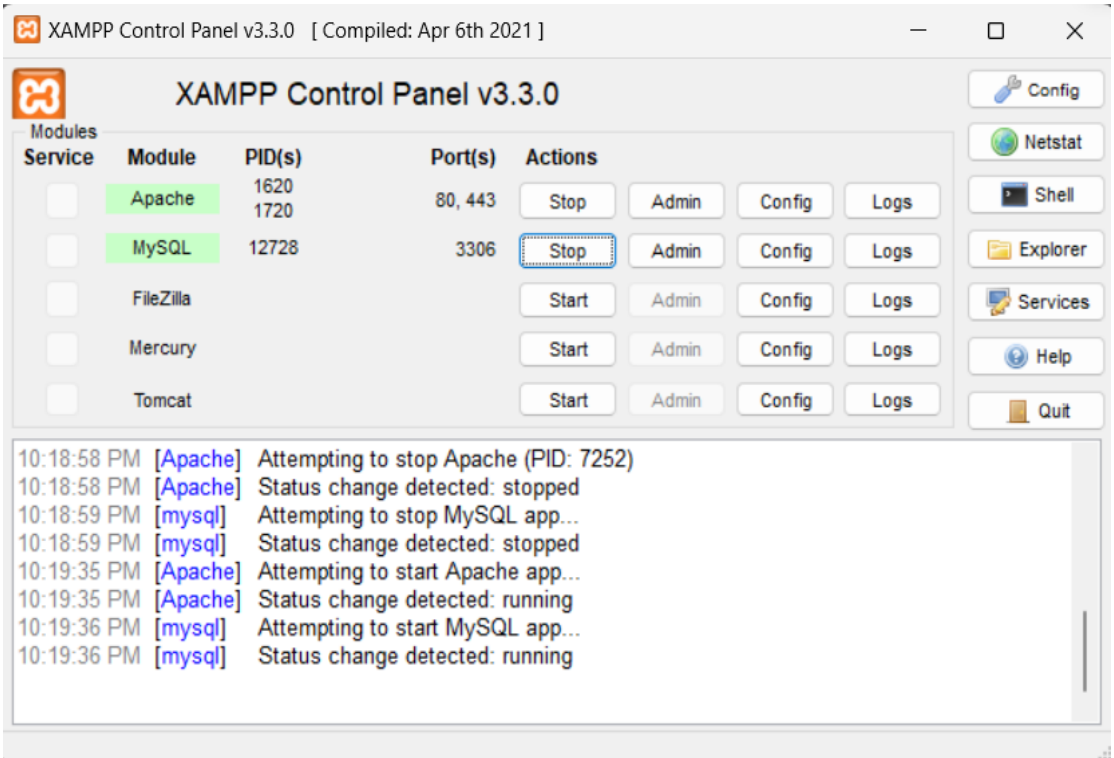
Laravel is a free, open-source PHP web framework designed for the development of robust and maintainable web applications. It follows the Model-View-Controller (MVC) architectural pattern and provides elegant syntax and tools for tasks such as routing, caching, database interactions, and more. Laravel aims to make the development process enjoyable and efficient by emphasizing clean, expressive code.

VII. MySQL:

MySQL is an open-source relational database management system (RDBMS) that is widely used for building and managing databases. It is a key component of the LAMP (Linux, Apache, MySQL, PHP/Python/Perl) and MERN (MongoDB, Express.js, React, Node.js) stacks and is often used in conjunction with web applications. MySQL is known for its performance, reliability, ease of use, and strong community support.

VIII. Xampp

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.



c. Goal and Purposed system

1. **Facilitating Transactions:** Resell Hub aims to provide a user-friendly platform that makes it easy for people to buy and sell a wide range of products and services. The platform acts as a mediator, allowing users to connect and complete transactions.
2. **Creating a Global Marketplace:** It operates in multiple countries, and its goal is to create a global marketplace where users from different regions can interact and trade goods. This helps in expanding the reach and diversity of the platform.
3. **Promoting User Engagement:** It's strive to keep users engaged on its platform. This involves providing features that enhance the user experience, such as easy navigation, robust search capabilities, and secure transaction processes.
4. **Building Trust and Safety**: Trust is crucial in online marketplaces. It's aim to build trust among its users by implementing safety measures, verifying user identities, and providing guidelines for secure transactions. This helps in reducing fraud and enhancing the overall safety of the platform.
5. **Empowering Individuals and Small Businesses:** This provides a platform for individuals and small businesses to reach a larger audience and sell their products or services. It aims to empower users by giving them the tools and visibility needed to succeed in online commerce.
6. **Monetization and Business Growth**: As a business, OLX aims to generate revenue through various monetization strategies, such as premium listings, advertising, and other value-added services. The ultimate goal is to achieve sustainable growth and profitability.

SYSTEM DESIGN

3. MODULE DIAGRAM

Module is a diagrammatic representation of a functionality within a project. It may have one or many module in a software. In other words, module is an approach that subdivides a system into smaller parts called Module that independently created and then in Different Systems. Each of a set of standardized parts or independent unit can be used to construct a more complex structure.

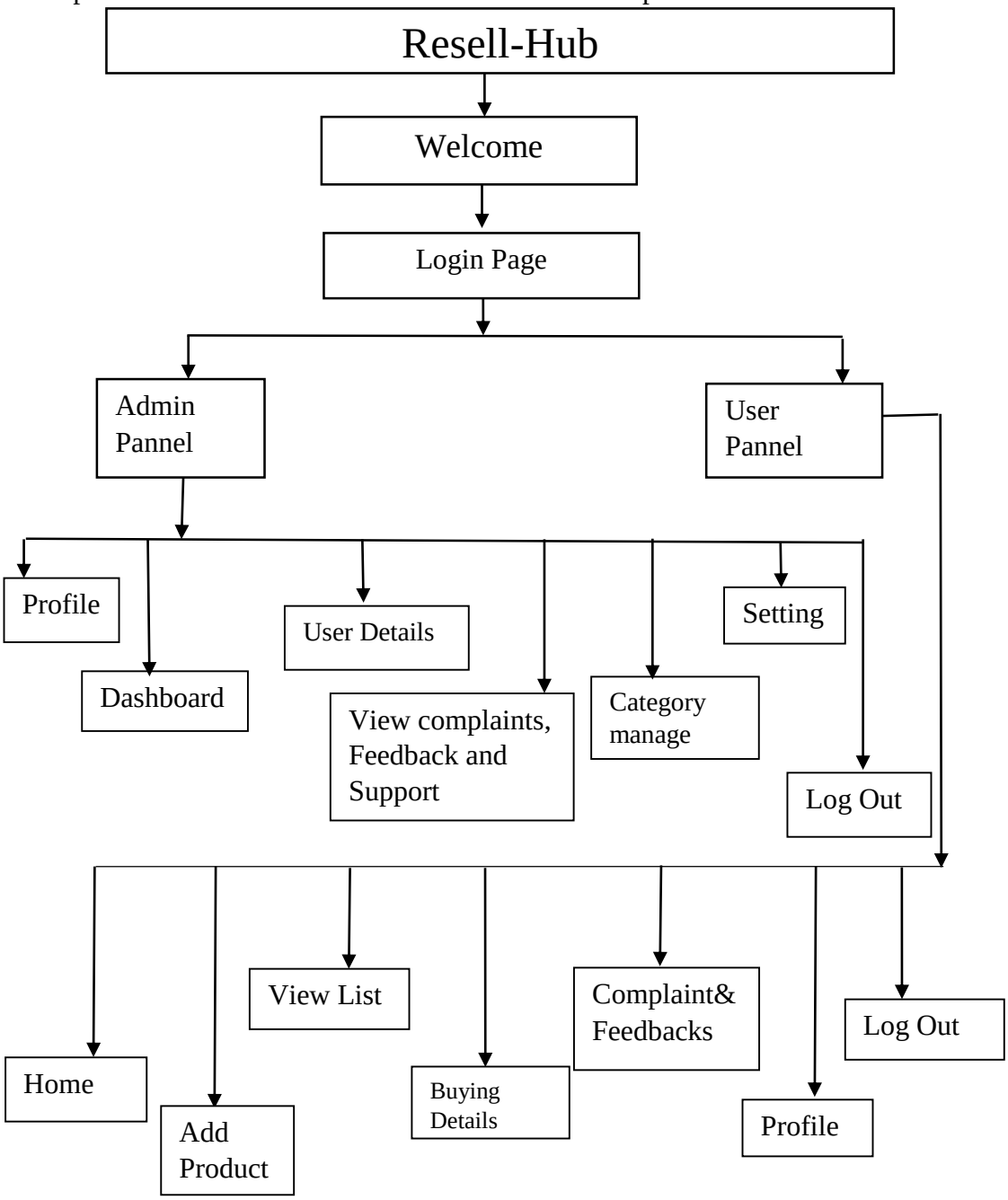


Fig-1 : Module Diagram

4. FUNCTION DIAGRAM

To explain how a software design may be represented as a set of functions which share system state information. To develop an example which illustrates the process of function oriented diagram. To compare using a common examples, sequential and concurrent function oriented diagram.

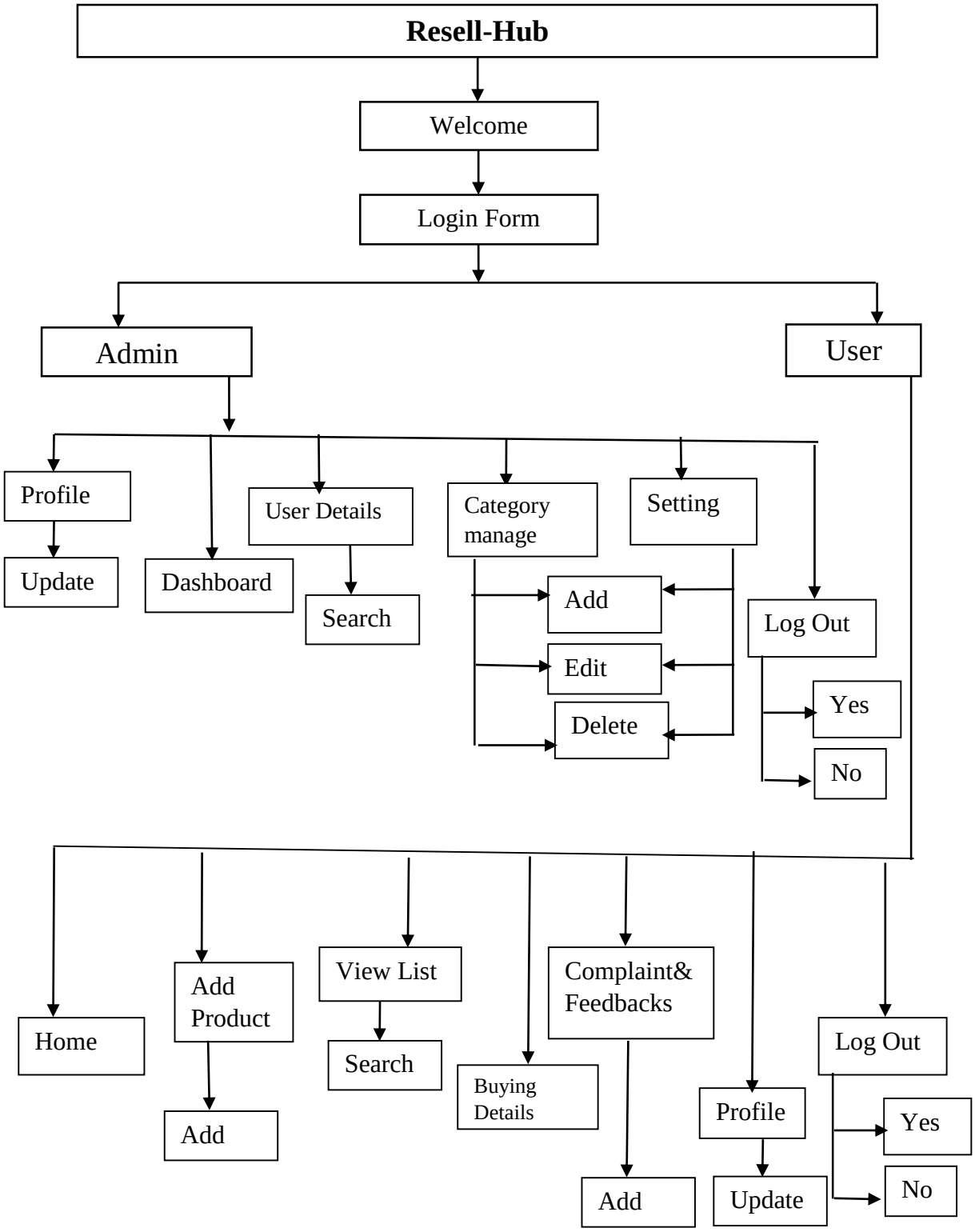
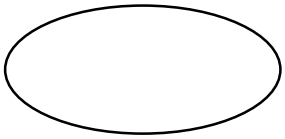
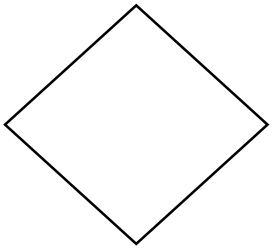
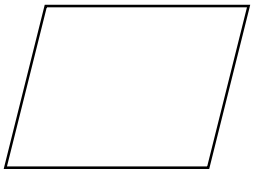
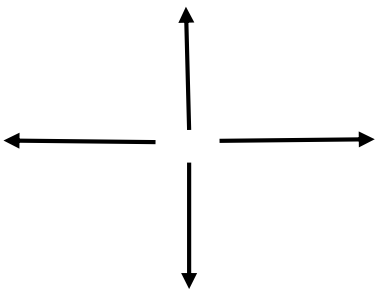


Fig-2 : Function Diagram

5. FLOWCHART

Flowchart can be defined as the geographical representation of an algorithm. The flowchart is an easy way to understand and analyze the problem. Flowchart uses some geometric figures to specify a particular operation. They are given below with their functions.

Geometric Figure	Names with function
	(Oval/Ellipse):- It is used as starting and stopping of the flowchart
	(Diamond):- It is used to decision making Process.
	(Rectangle):- It is used to process the data.
	(Parallelogram):- It is used to accept input data and Display the output result.
	(Flow Lines):- It is used as connectors.

a. Flowchart Diagram of Login Form :

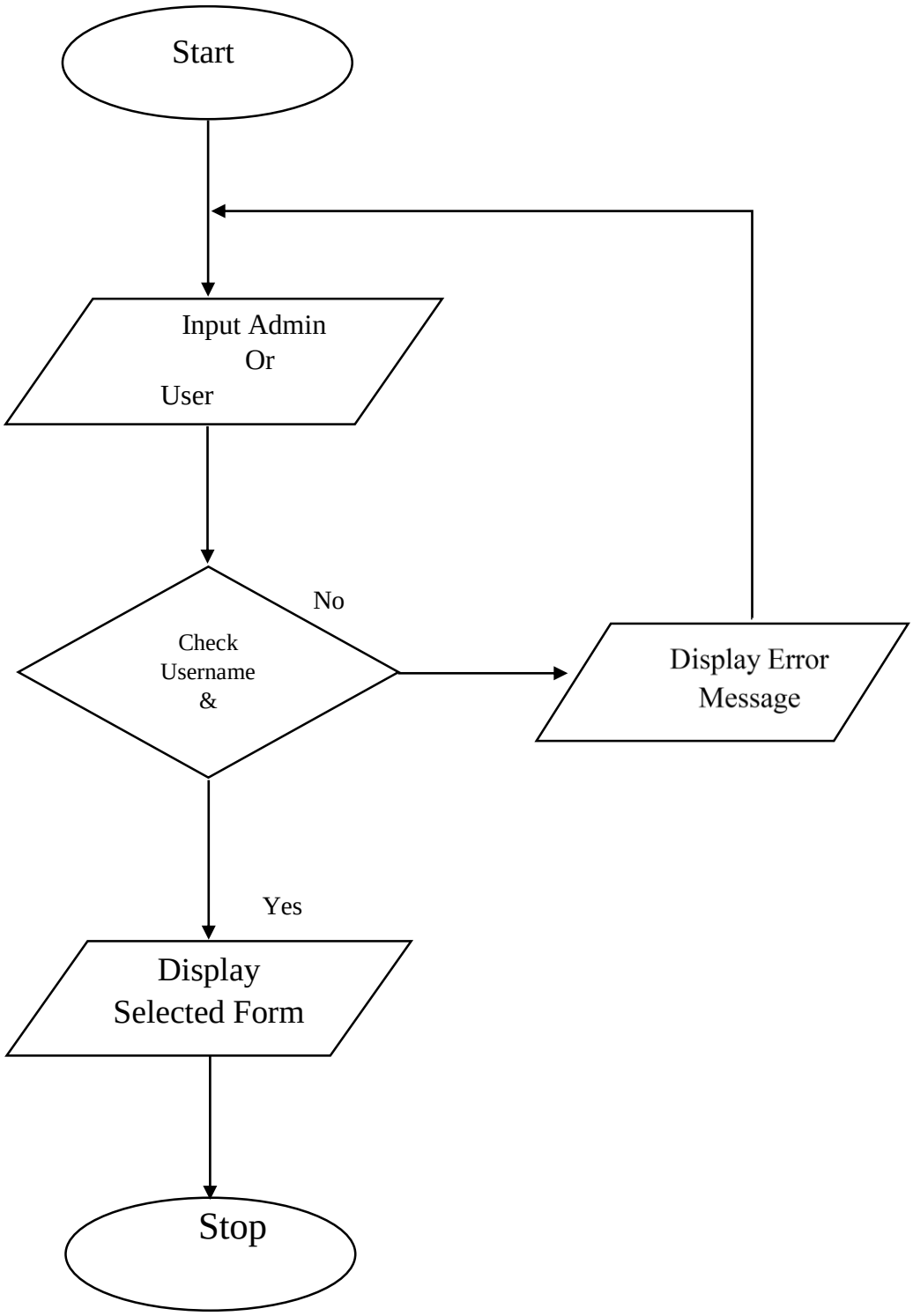


Fig-3.1: Flowchart of Login Form

b. Flowchart Diagram of Register Form :

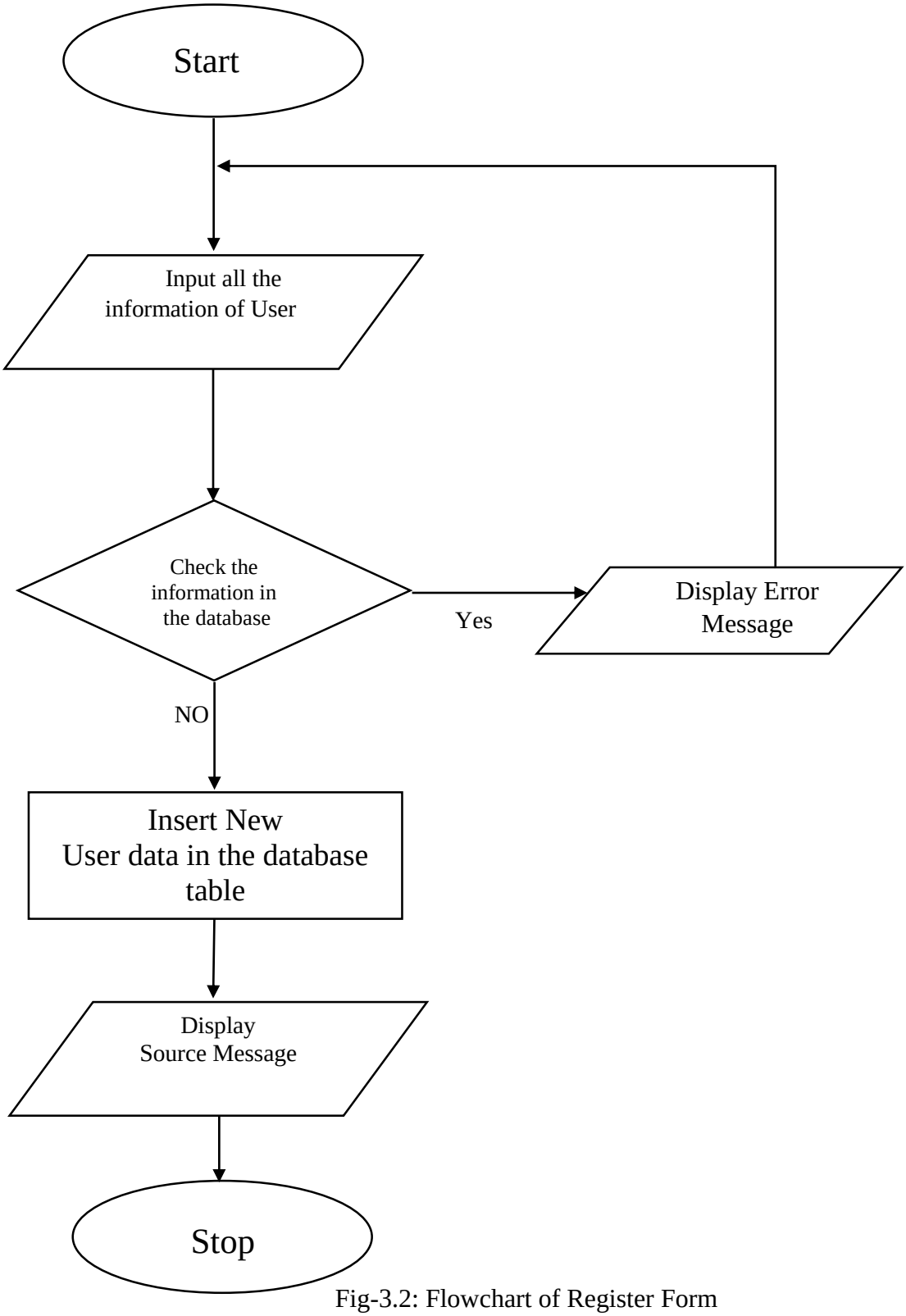


Fig-3.2: Flowchart of Register Form

c. Flowchart Diagram of Product Detail Form :

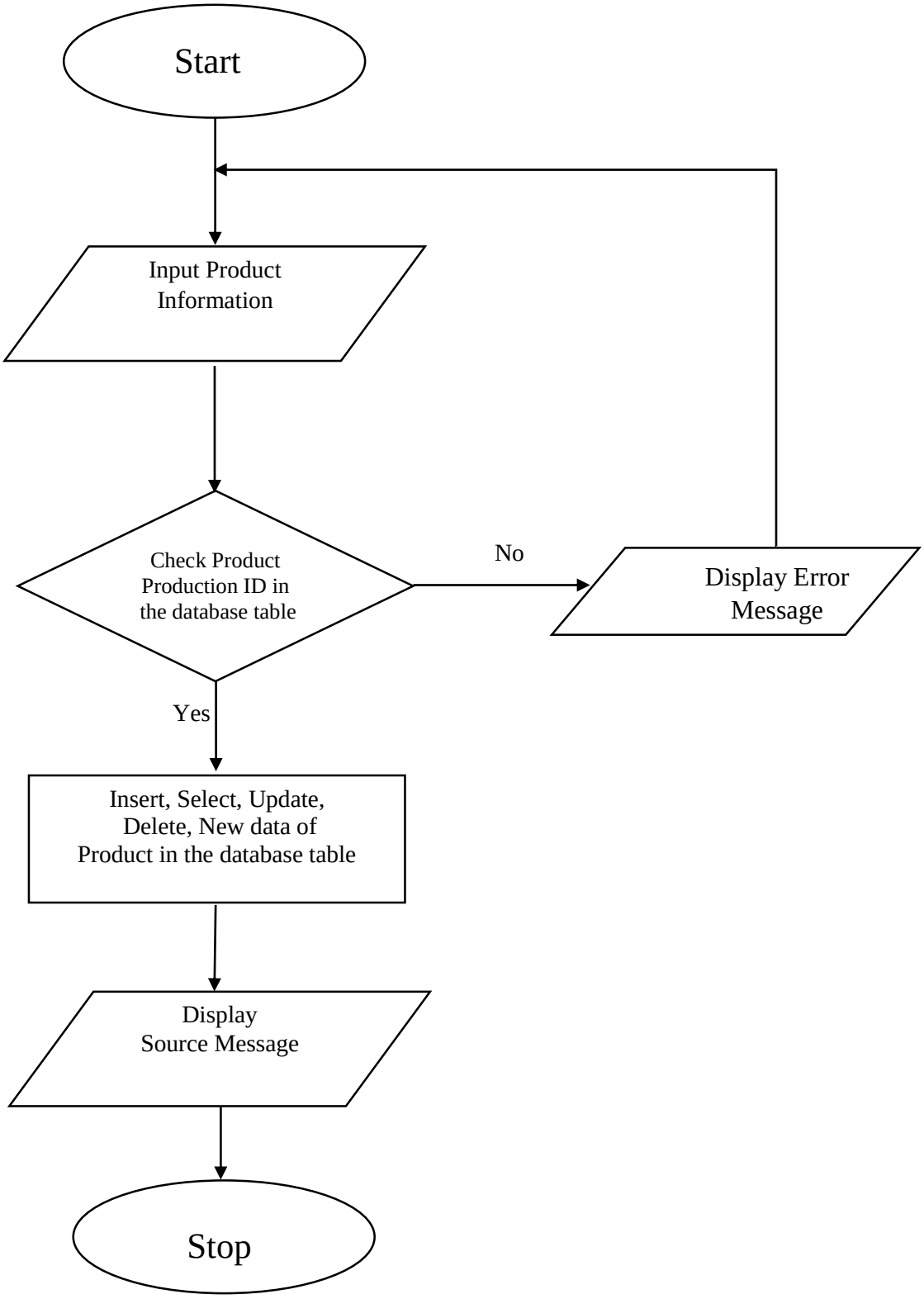


Fig-3.3: Flowchart of Product Details Form

d. Flowchart Diagram of Category Detail Form :

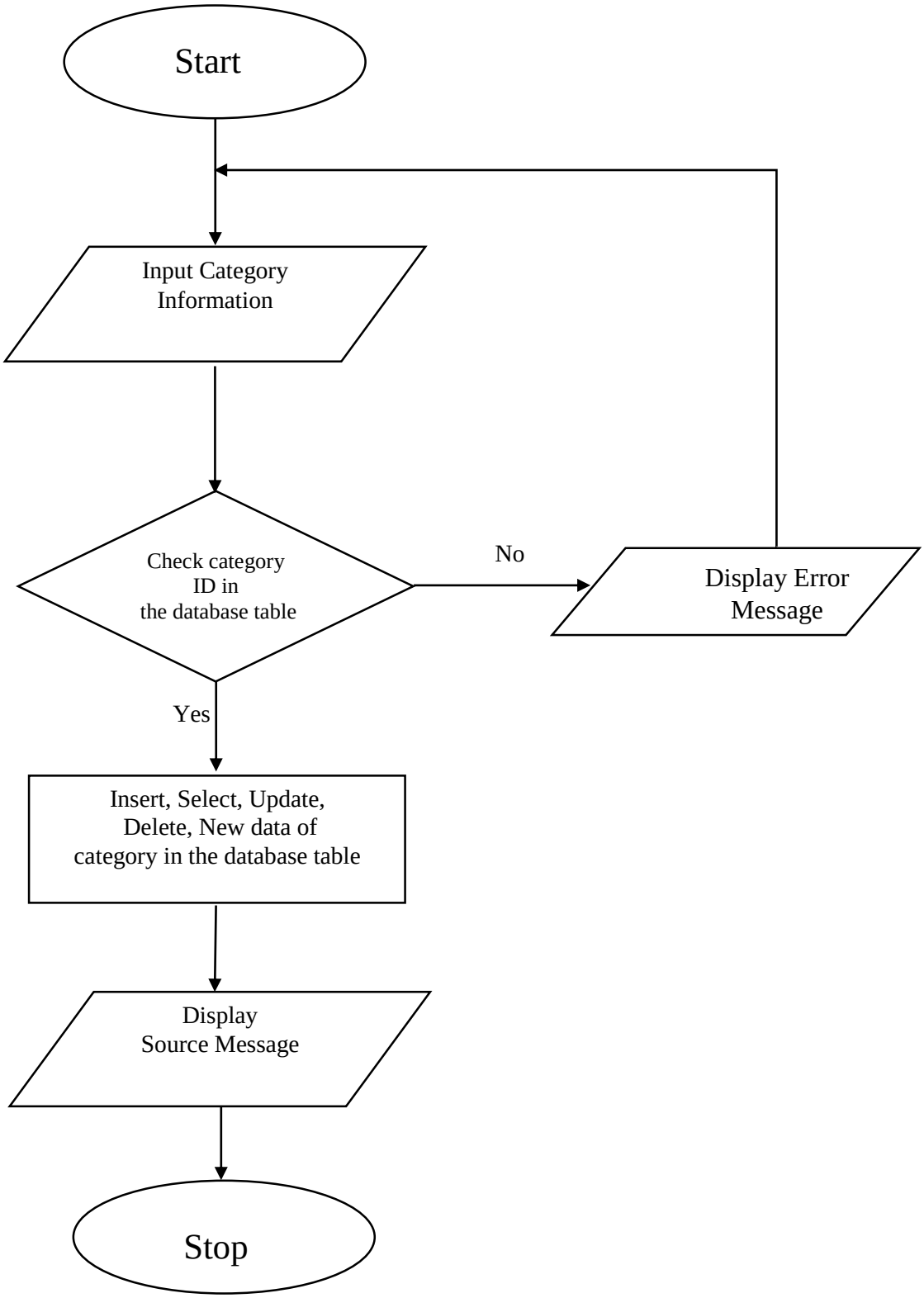


Fig-3.4: Flowchart of Category Details Form

e. Flowchart Diagram of Search Form :

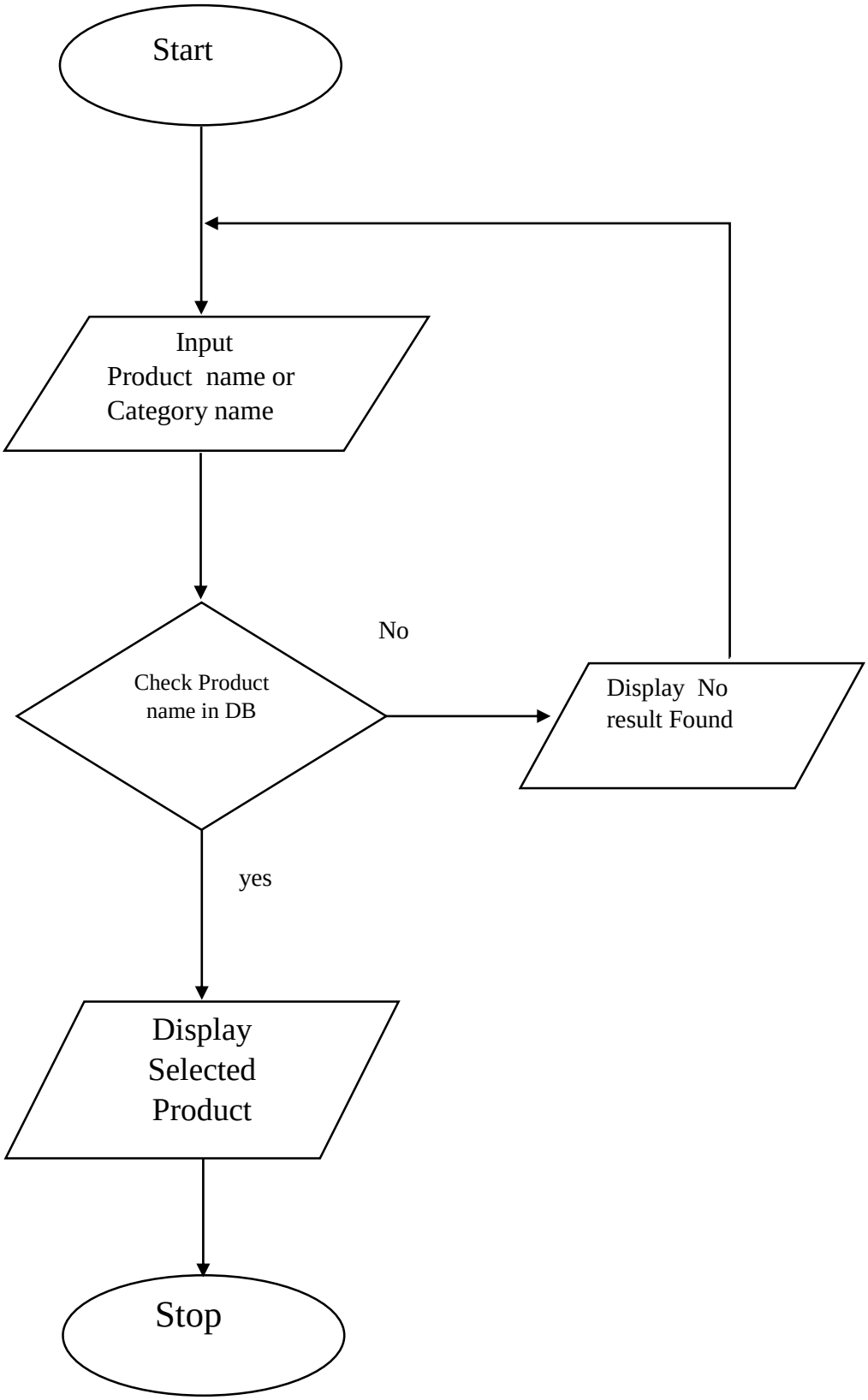
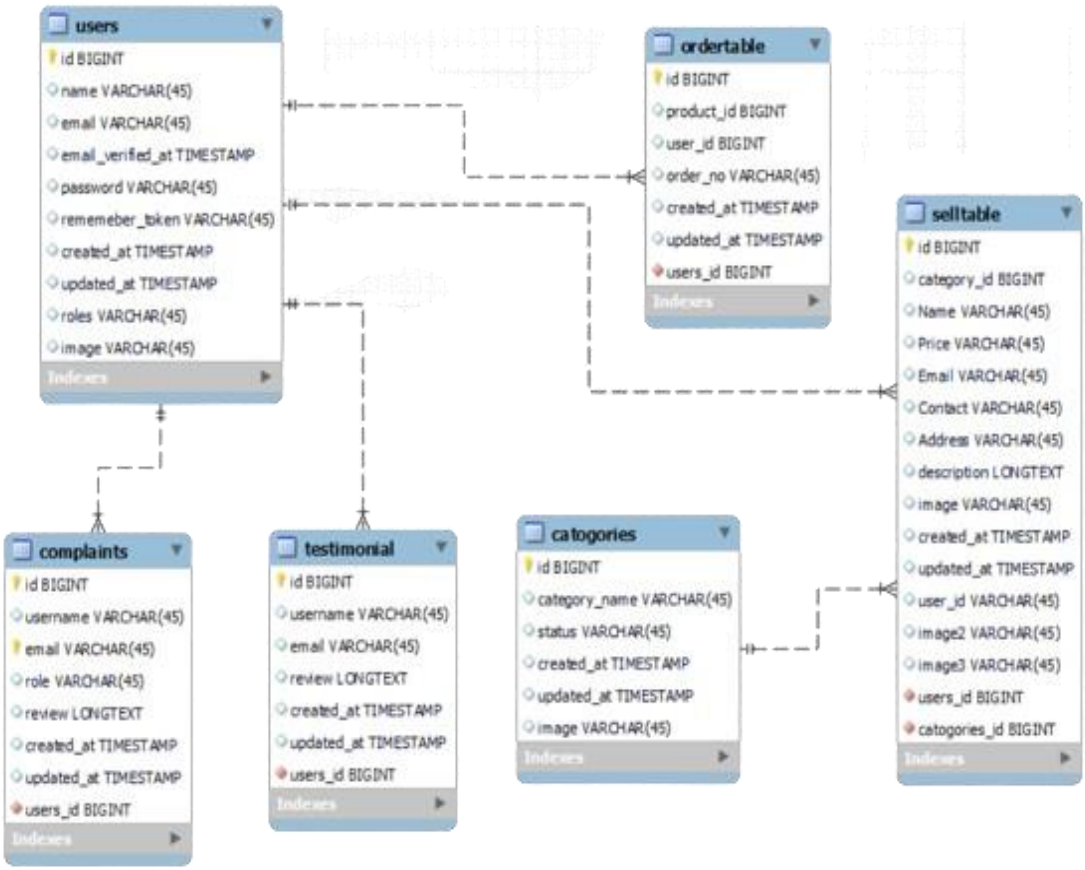


Fig-3.5: Flowchart of Search Form

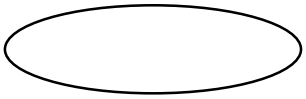
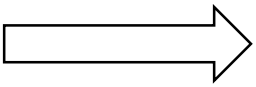
6. ER-DIAGRAM

An entity relationship diagram is a data modeling techniques that graphically illustrate an information system and the relationships between those entities. An ER- Diagram is a conceptual and sensation model of data used to represents entity framework infrastructure.



7. DATAFLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. It is a tool for visualizing the movement of data through a system, and for identifying and understanding the relationships between different components of the system.

Geometric Figure	Names	Function
	Rectangle	A rectangle defines source or destination of the system. It is also called entity.
	Oval	It represents as a process that gives us information. It is also called processing box.
	Arrow	An arrow identifies the data flow i.e. it gives information to the data that is in motion.
	Open Rectangle	Data is store either temporary or permanently.

a. 0-level DFD:

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

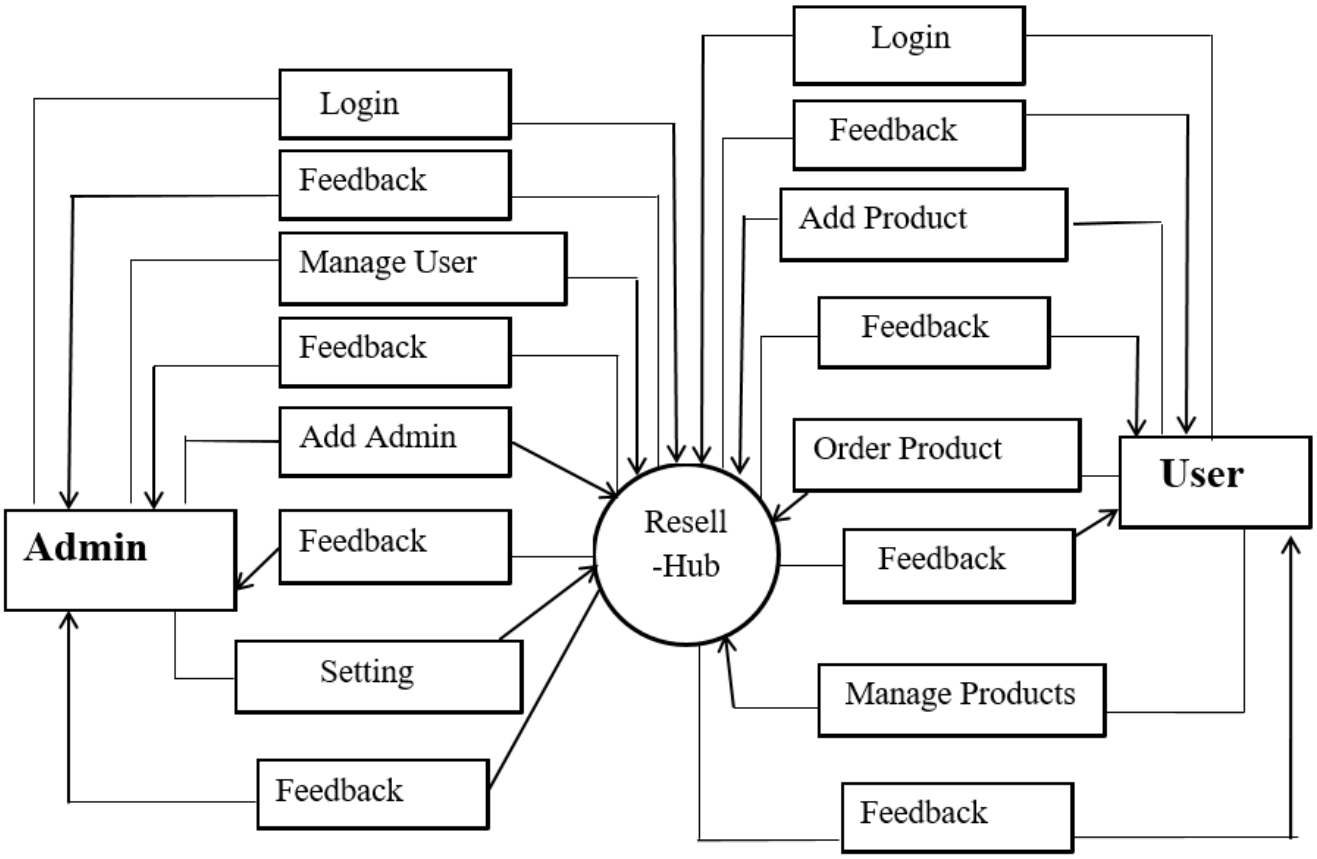


Fig 5.1.0-level DFD

b. DFD of Admin Login System

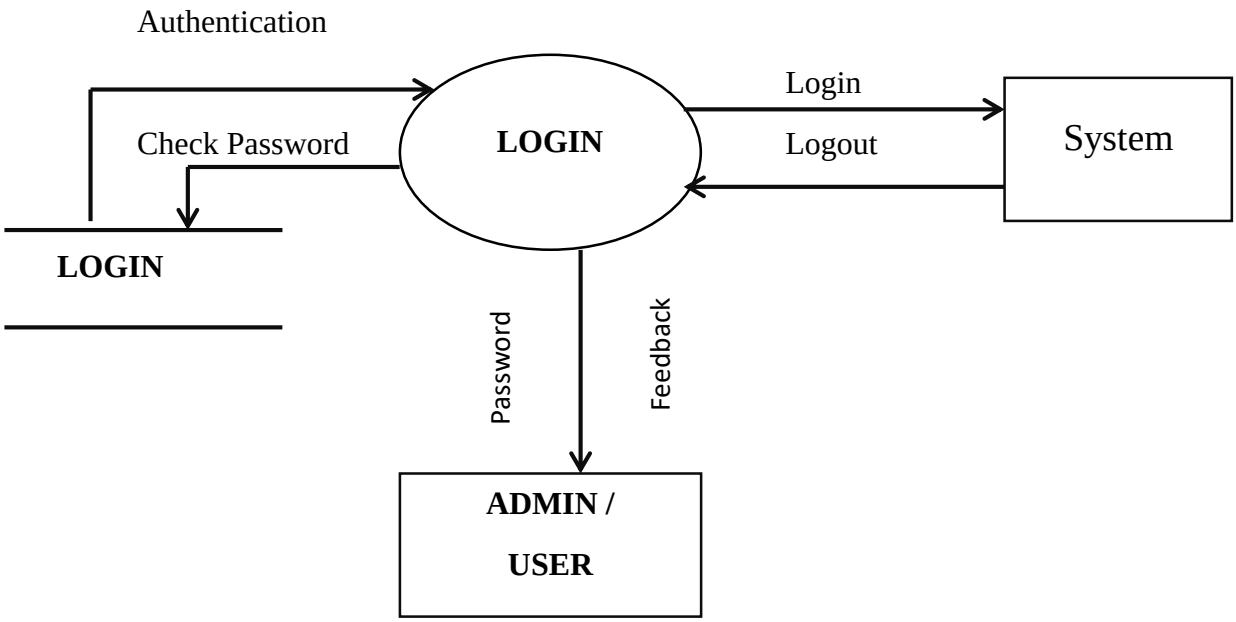


Fig 5.2. DFD of Login System

c. DFD of Change Password

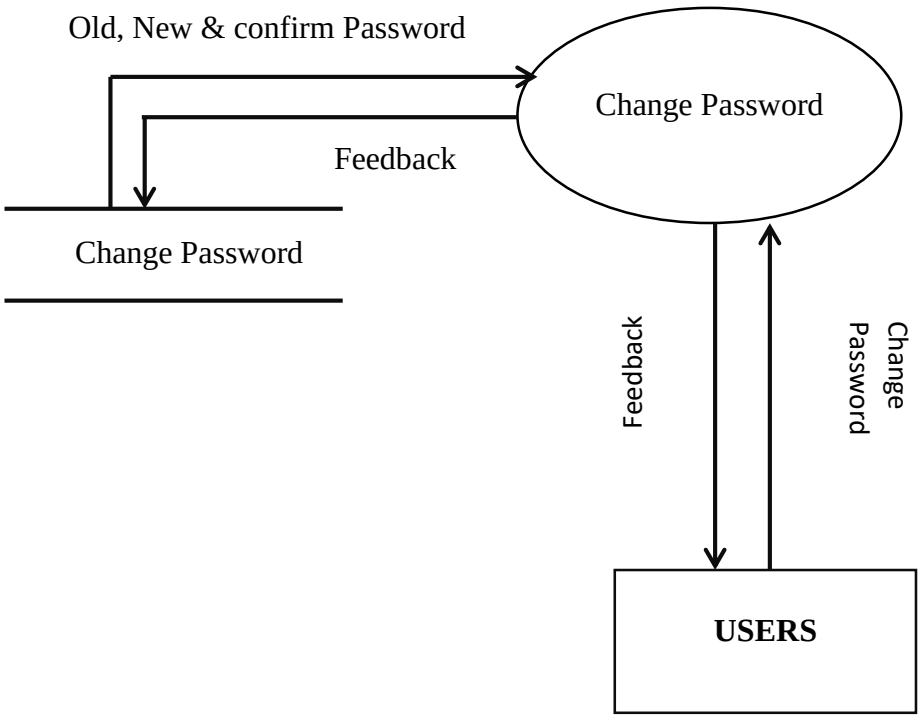
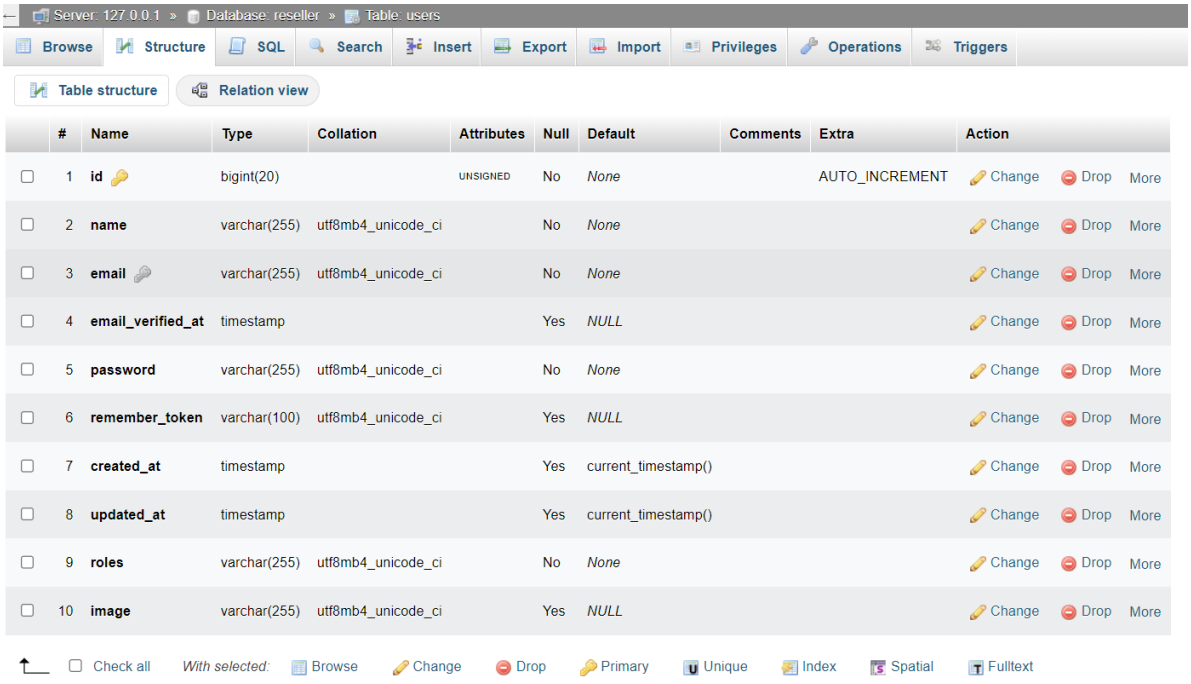


Fig 5.3. DFD of Change Password

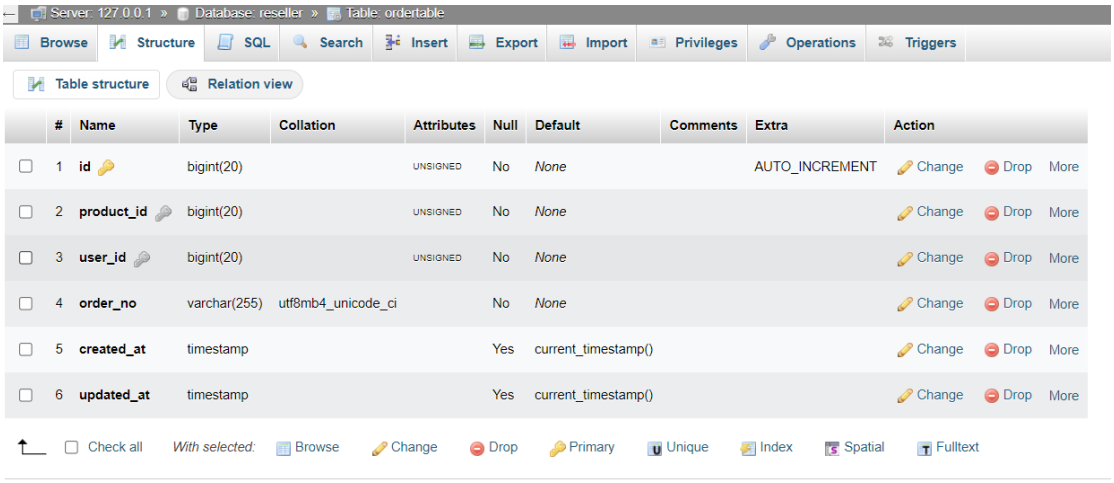
8. Database Diagram

a. User Table



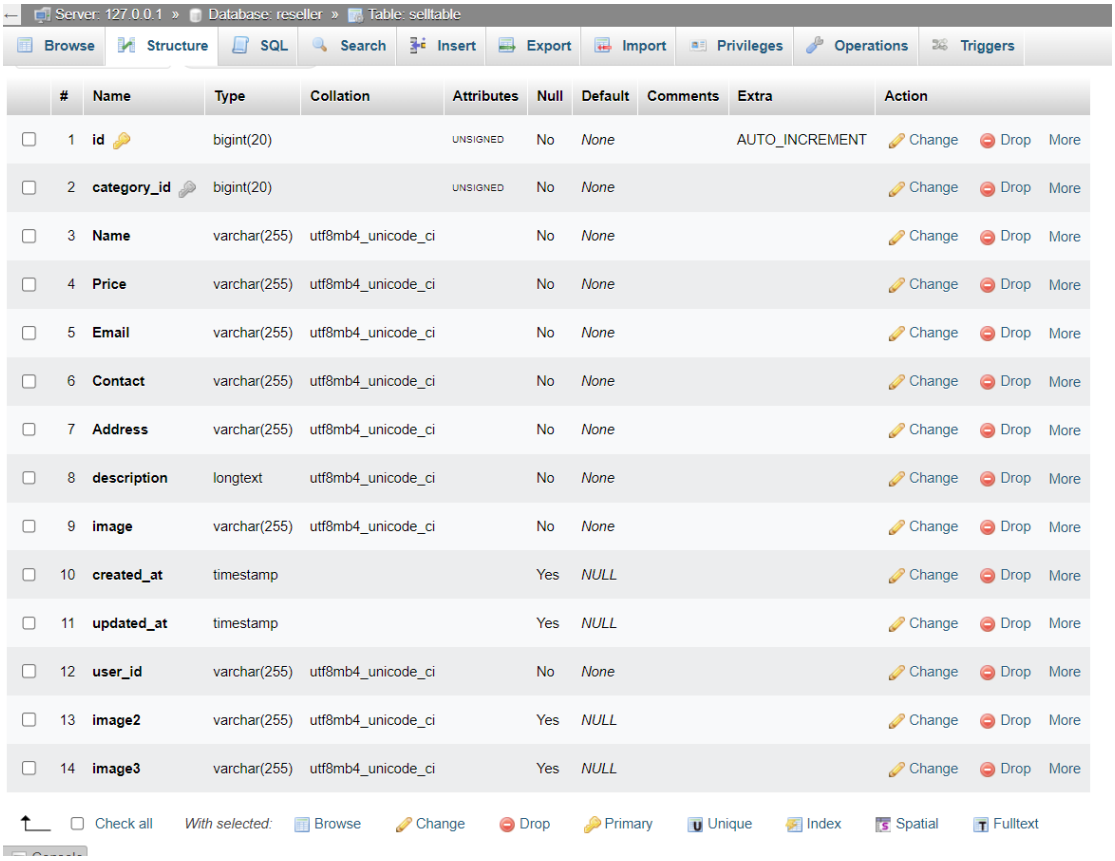
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	email_verified_at	timestamp			Yes	NULL			Change Drop More
5	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
7	created_at	timestamp			Yes	current_timestamp()			Change Drop More
8	updated_at	timestamp			Yes	current_timestamp()			Change Drop More
9	roles	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
10	image	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More

b. Order Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	product_id	bigint(20)		UNSIGNED	No	None			Change Drop More
3	user_id	bigint(20)		UNSIGNED	No	None			Change Drop More
4	order_no	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	created_at	timestamp			Yes	current_timestamp()			Change Drop More
6	updated_at	timestamp			Yes	current_timestamp()			Change Drop More

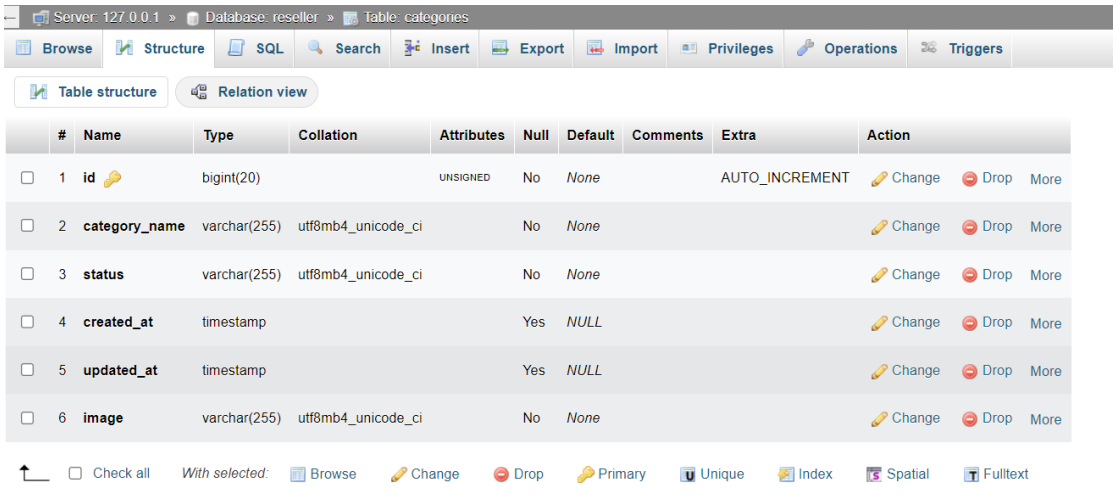
c. Sell Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	category_id	bigint(20)		UNSIGNED	No	None			Change Drop More
3	Name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	Price	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	Email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	Contact	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
7	Address	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
8	description	longtext	utf8mb4_unicode_ci		No	None			Change Drop More
9	image	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
10	created_at	timestamp			Yes	NULL			Change Drop More
11	updated_at	timestamp			Yes	NULL			Change Drop More
12	user_id	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
13	image2	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
14	image3	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

d. Categories Table



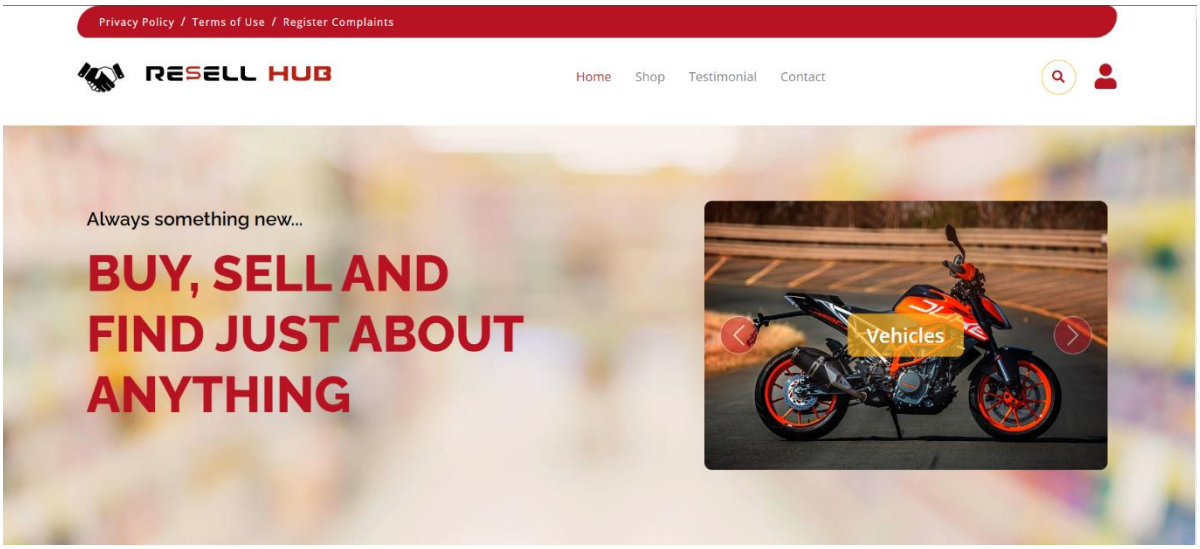
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	category_name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	status	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	created_at	timestamp			Yes	NULL			Change Drop More
5	updated_at	timestamp			Yes	NULL			Change Drop More
6	image	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

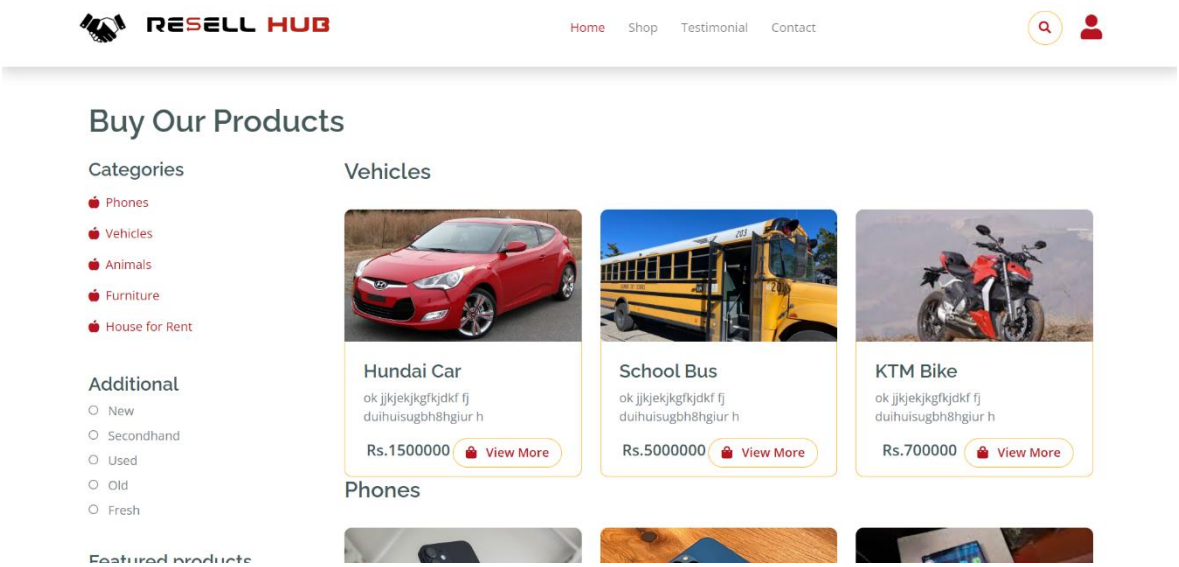
USER INTERFACE

9. User Interface

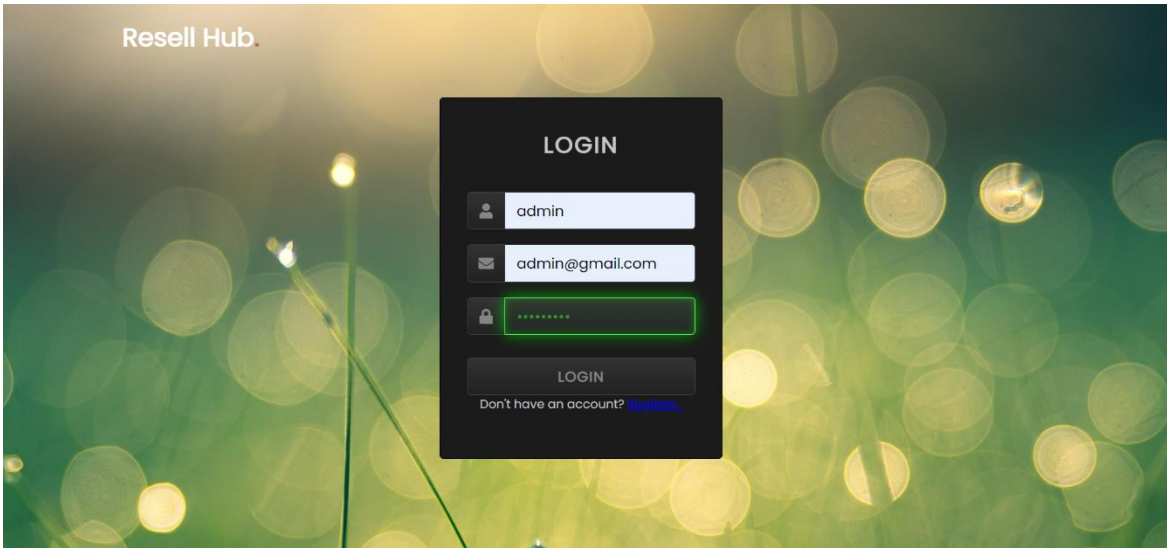
9.1 Welcome



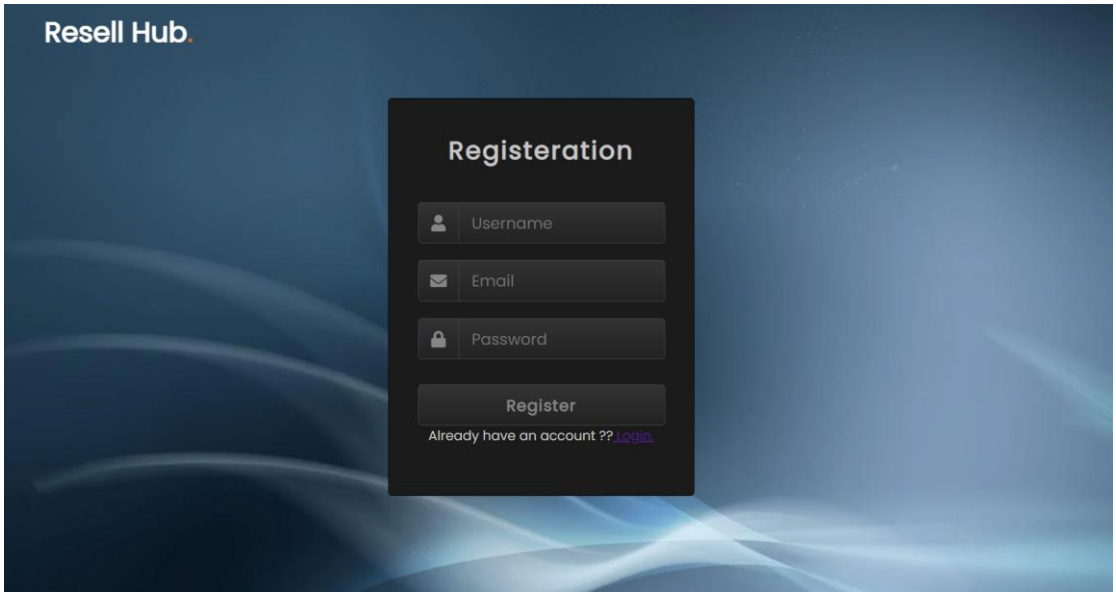
9.2 Shop



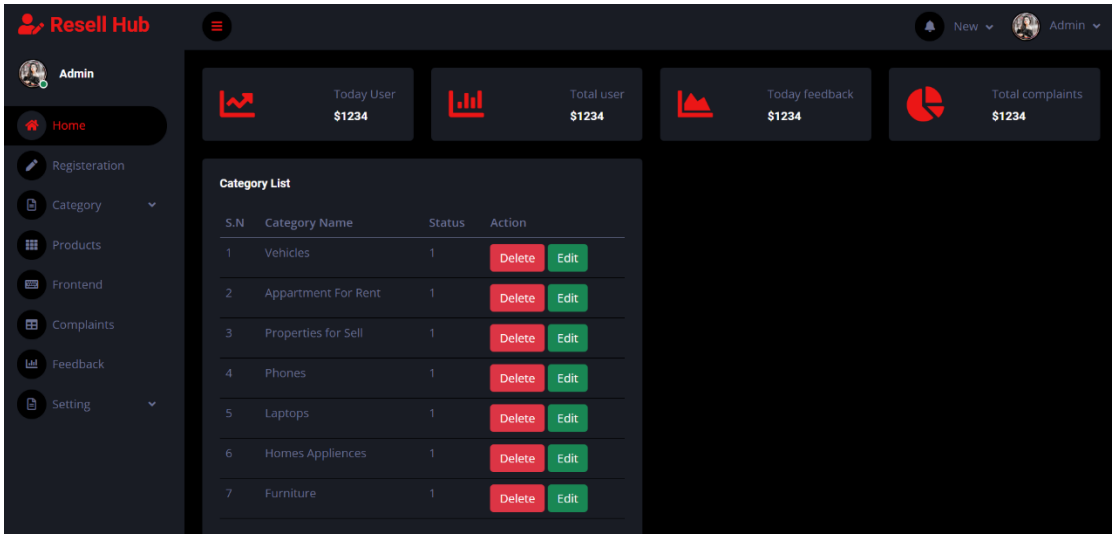
9.3 Login



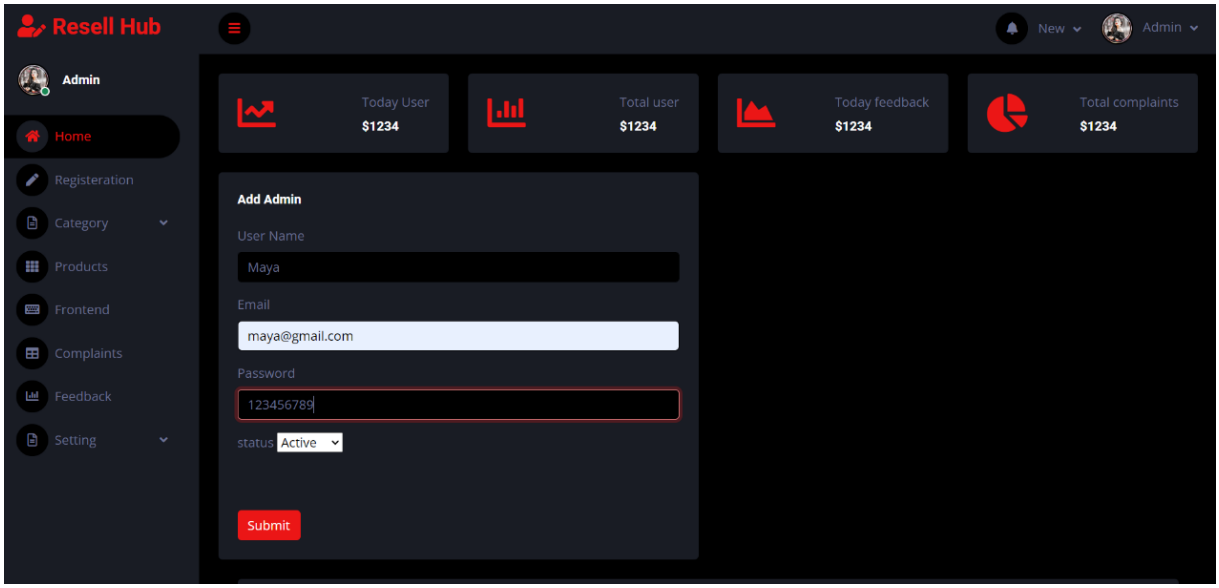
9.4 Register



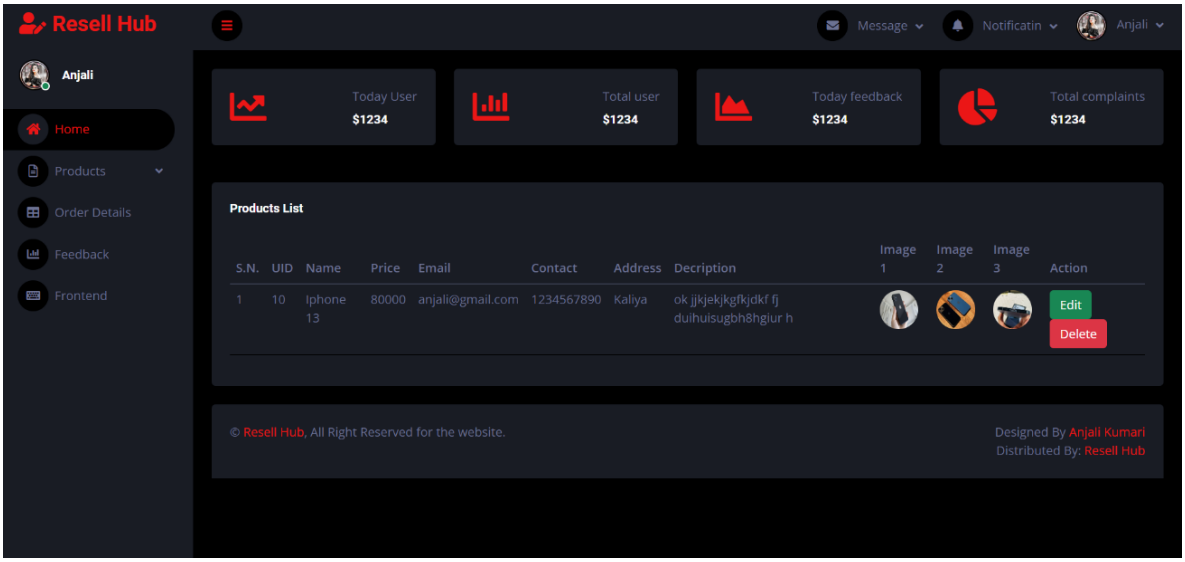
9.5 Admin Panel



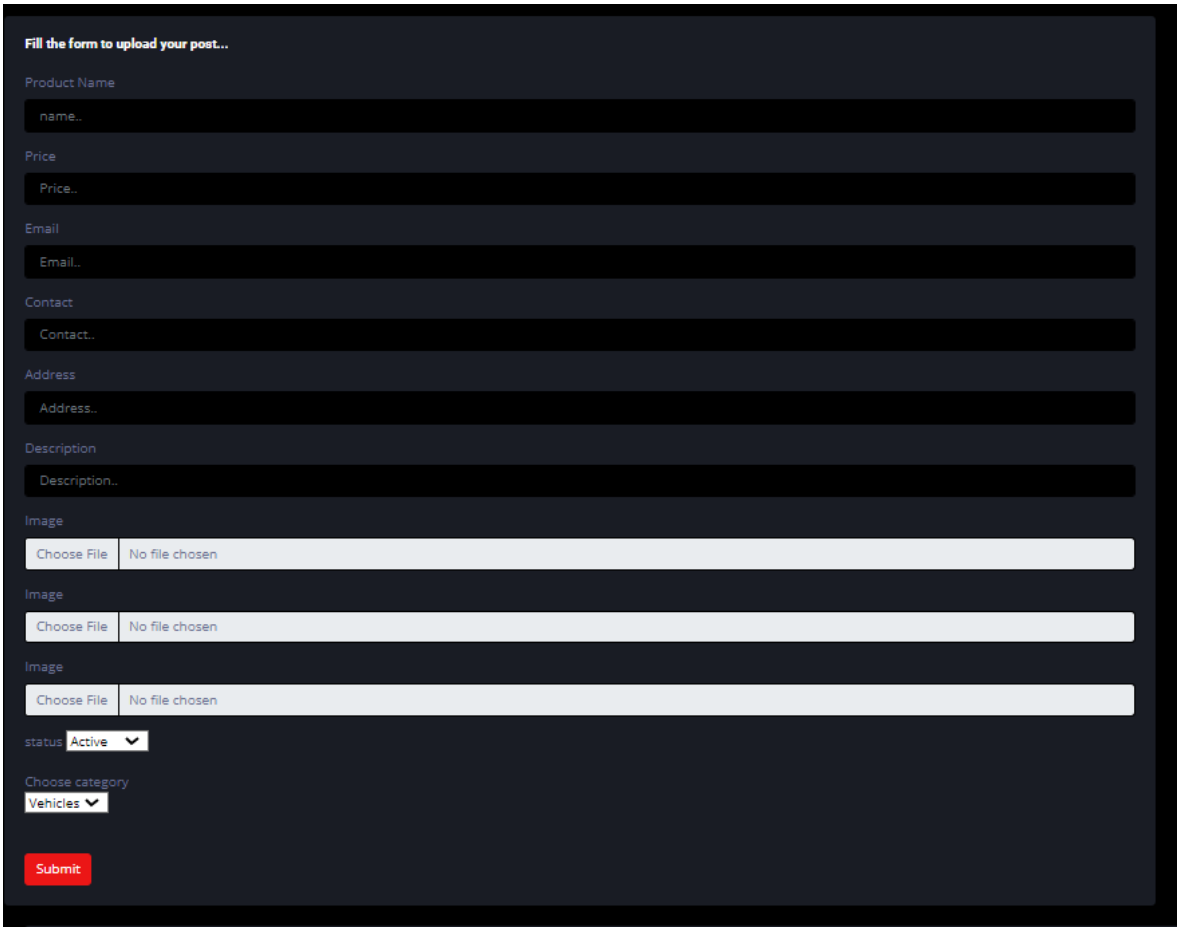
9.6 Add Admin



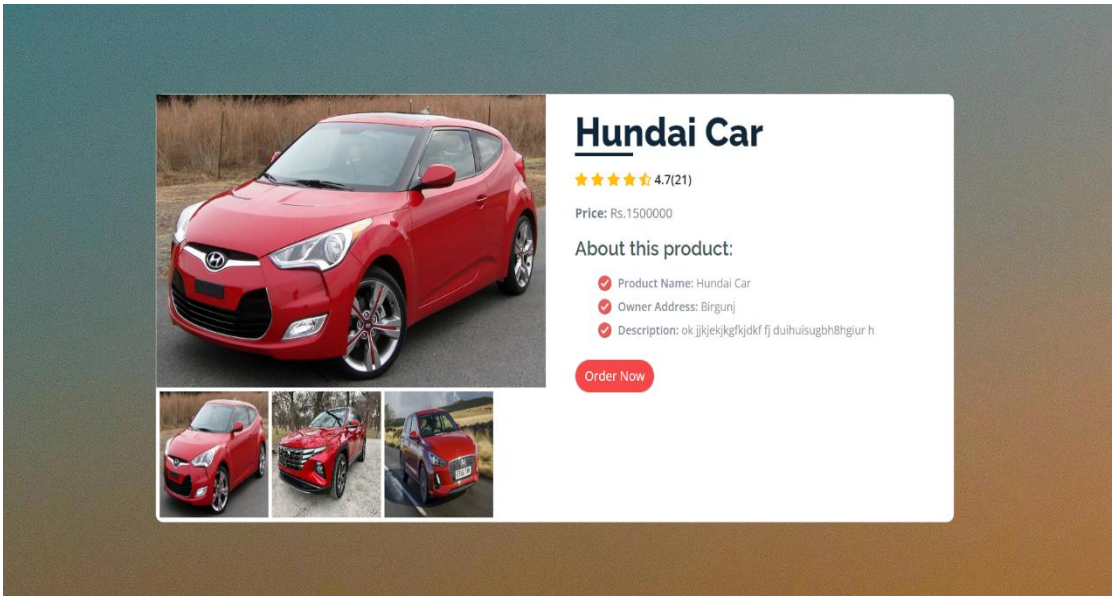
9.7 User Panel



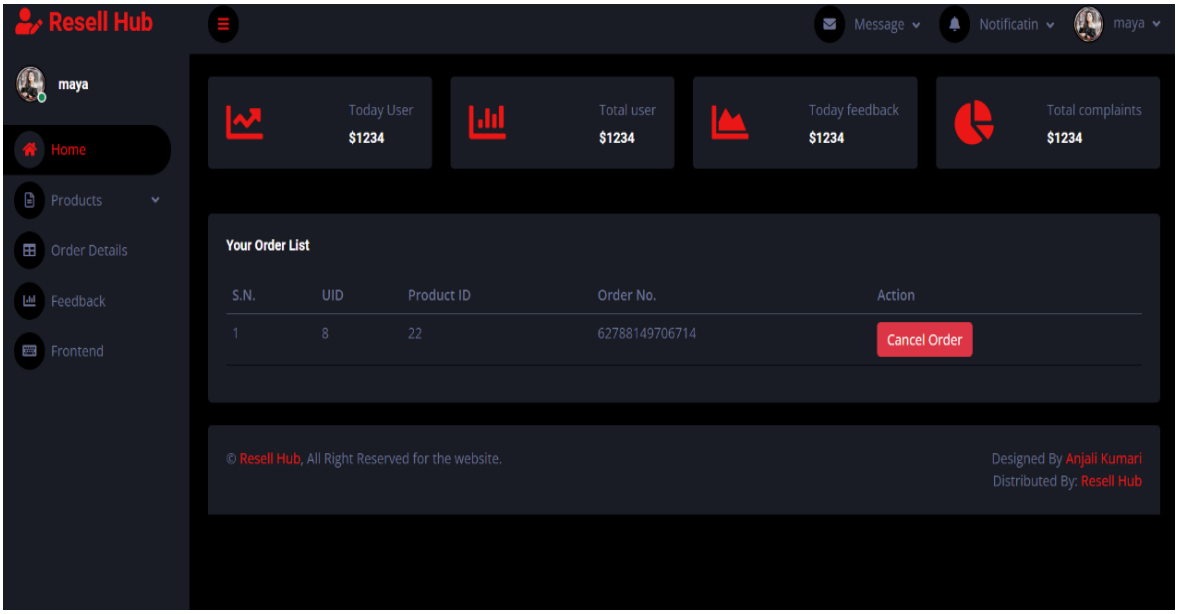
9.8 Add Product



9.9 View More



9.10 Order Details



CODING

10.1 Code For Login

```php
public function login(Request $request){
        // dd($request->all());

        $request->validate([
            'email' => 'required | email',
            'password' => 'required | min:6'
        ]);

        $email = $request->email;
        $password = $request->password;
        $user = User::where('email',$request->email)->first(); //finding user

        // dd($user);
        if($user){
            if($user->password){
                if(Hash::check($request->password, $user->password)){
                    Auth::login($user);
                      $userRoles = Auth::user()->roles;
                      $userid = Auth::user()->id;
                      $request = session()->put('id',Auth::user()->id);

                    switch ($userRoles) {
                        case ($userRoles == 'Admin' ):
                            return redirect()->route('admin.dashboard');
                            break;

                        case ($userRoles == 'User'):
                            return redirect()->route('User.dashboard');
                            break;

                        default:
                            return redirect()->back();
                    }
                }
                $request->session()->flash('error','Incorrect Password');
                return redirect()->back();
            }
        }
        $request->session()->flash('error','User Not Found');
        return redirect()->back();
    }
```

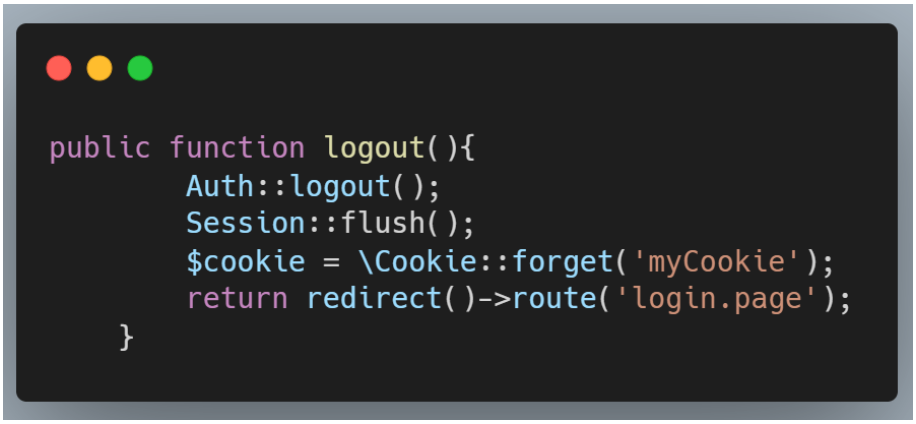
10.2 Code For Register

```php
public function register(Request $request){

    $request->validate([
            'name'=>'required',
            'email' => 'required | email | unique:users',
            'password' => 'required | min:6'
        ]);

    $data = [
        'name'=>$request->name,
        'email'=> $request->email,
        'password'=>bcrypt($request->password),
        'roles'=> 'User'

   ];

//    dd($data);
   user::insert($data);

   return redirect()->route('login.page');
    }
```

10.3 Code For Display data

```php
//Display table  Data in UI card//
    public function displayproduct(){
        $id = session()->get('id');
        $data= sellmodel::where('user_id',$id)->get();
        return
view}'backend.User.Products.ManageProduct',compact('data'));
```

10.4 Code For Log Out



```
public function logout(){
        Auth::logout();
        Session::flush();
        $cookie = \Cookie::forget('myCookie');
        return redirect()->route('login.page');
    }
```

10.5 Code For Order Detail

```
public function order(Request $request,$id){
       // dd($id);
       if($user = Auth::id()){
           $product = sellmodel::find($id);
               $P_id = $product->id;
               $usrId = $product->user_id;
               $orderNo = rand(10,999999999999999);

               $data =[
                   'order_no' => $orderNo,
                   'product_id'=>$P_id,
                   'user_id'=>Auth::id()
               ];

               // dd($data);
               OrderModel::insert($data);
               Mail::to(\auth()->user()->email)->send(new OrderMail($orderNo, $product));
               return redirect()->route('User.dashboard');
       }else{
           return redirect('login.page');
       }
   }
```

10.6 Code for create with multiple images

```
public function sellcreate(Request $request){
        // dd($request);

        $request->validate([
            'category_id'=>'required',
            'name' => 'required',
            'price' =>'required',
            'email' => 'required | email ',
            'contact' => 'required',
            'address' => 'required',
            'description' => 'required',
            'image' => 'required',
        ]);


        $images = array();

        if($files = $request->file('image') && is_array($request->file('image'))){

            for ($i = 0; $i <= 2; $i++) {
                $imageKey = 'image' . $i;
                if ($request->file('image')[$i]) {
                    $file =  $request->file('image')[$i];
                    $newName = time().'_'. rand(10,9999999999999).'_'.$file-
>getClientOriginalName();
                    $newPath = public_path()."/Product_Images/";
                    $file->move($newPath, $newName);
                    $images[$imageKey] = $newName;
                }
            }

            // dd($images);

            $data = [
                'category_id'=>$request->category_id,
                'Name'=>$request->name,
                'Price'=>$request->price,
                'Email'=>$request->email,
                'Contact'=>$request->contact,
                'Address'=>$request->address,
                'description'=>$request->description,
                'user_id'=>$request->session()->get('id'),
                'image' =>$images['image0'] ?? "",
                'image2'=>$images['image1'] ?? "",
                'image3'=>$images['image2'] ?? "",
            ];

    //    dd($data);
            sellmodel::insert($data);
            return redirect()->back();
    }
}

    public function displayData(){
        $data['categories']= category::where('status',1)->get();
         return view('backend.User.Products.Sell',$data);

}
```

10.7Code For Edit

```
//Edit Data
   public function edit($id)
   {
    if(!$id){
        return redirect()->back();
    }
    $Product_data= sellmodel::find($id);
   if($product_data){
    return
view('backend.User.EditProduct',compact('product_data'));
   return redirect()->back();
   }
```

10.8Code For Update

```
//Update Data
   public function update(Request $request,$id)
   {
    if(!$id){
        return redirect()->back();
    }
    $product_data= sellmodel::find($id);
   if($product_data){
    $data=[
        'category_id'=>$request->category_id,
                'Name'=>$request->name,
                'Price'=>$request->price,
                'Email'=>$request->email,
                'Contact'=>$request->contact,
                'Address'=>$request->address,
                'description'=>$request->description,
                'user_id'=>$request->session()->get('id'),
                'image' =>$images['image0'] ?? "",
                'image2'=>$images['image1'] ?? "",
                'image3'=>$images['image2'] ?? "",
            ];
   $product_data->update($data);
  return redirect()->route('manage-product');

   }
   return redirect()->back();
}
```

10.9 Code For Delete

```
//Delete Record
    public function delete($id){
        if(!$id){
            return redirect()->back();
        }

       $cat_data= category::find($id);
       if($cat_data){
        $cat_data->delete();
       }
       return redirect()->back();
       }
```

SYSTEM TESTING

11. System Testing

- System testing for our Resell-Hub Application involves evaluating the integrated system to ensure that it meets the specified requirements and functions as intended. Here is an overview of key aspects that are considered in the system testing phase:

I. **Functional Testing:** Verify that all functional requirements are implemented correctly. Test user registration and authentication processes. Ensure the virtual tour feature works seamlessly.

II. **User Interface (UI) Testing:** Evaluate the user interface for usability, responsiveness, and adherence to design specifications. Verify that buttons, links, and interactive elements function as expected.

III. **Performance Testing:** Assess the performance of the application under various conditions. Measure response times for key functionalities.

IV. **Security Testing:** Identify and address security vulnerabilities to protect user data. Verify secure communication, especially during payment transactions.

V. **Compatibility Testing:** Confirm that the Resell-Hub Application works correctly across different browsers, devices, and operating systems. Verify compatibility with various devices, including desktops, laptops, tablets, and smartphones.

VI. **Database Testing:** Ensure the integrity and reliability of the database. Test data retrieval and storage operations. Verify that database queries are optimized for efficiency. Check the handling of concurrent transactions and data consistency.

SYSTEM MAINTENANCE

12. System Testing

System maintenance is a crucial aspect of ensuring the ongoing functionality, security, and usability of the Roomie project. It involves monitoring, updating, and enhancing the system to meet changing requirements and address issues that may arise over time.

1. **Monitoring and Incident Response:**
    - Implement continuous monitoring to detect and address any system anomalies or security incidents promptly.
    - Establish an incident response plan to efficiently handle and resolve issues as they arise.
2. **Regular Software Updates:**
    - Keep the application's software stack, including operating systems, frameworks, and third-party libraries, up to date to address security vulnerabilities and benefit from the latest features and optimizations.
3. **Database Maintenance:**
    - Perform routine database maintenance tasks, such as optimizing queries, updating indexes, and managing data backups to ensure data integrity and efficient performance.
4. **User Feedback and Support:**
    - Establish channels for users to provide feedback on their experiences with the application.
    - Address user-reported issues promptly and provide ongoing support to maintain high user satisfaction.
5. **Scalability and Performance Optimization:**
    - Monitor system performance and scalability to anticipate and address any bottlenecks as user traffic grows.
    - Optimize code and database queries to enhance overall system performance.
6. **Regulatory Compliance:**
    - Stay informed about changes in relevant legal and regulatory requirements, ensuring ongoing compliance with data protection laws and other industry-specific regulations.

CONCLUSION

13. CONCLUSION:

The conclusion of the Resell-Hub project marks the culmination of extensive planning, development, and testing efforts aimed at creating a Resell-Hub Application that addresses the diverse needs of users in search of suitable accommodations. This concluding phase encapsulates the achievements, highlights the key features, and reinforces the project's overarching goals.

The project has successfully adopted a user-centric approach, ensuring that the Resell-Hub application is not only functional but also intuitive and responsive to the needs of its users. The incorporation of innovative features, such as the smart matching algorithm and immersive virtual tours, demonstrates a commitment to leveraging cutting-edge technology to enhance the overall user experience.

Transparent pricing, clear rental policies, and a robust verification process contribute to building trust among users, fostering a reliable and transparent housing marketplace. Users goes beyond a transactional platform, fostering community engagement through forums and networking features. This creates a collaborative environment and a sense of belonging among users.

In conclusion, the Roomie project not only represents the successful development of a Resell-Hub Application but also embodies a commitment to excellence, user satisfaction, and continuous improvement. As User is introduced to the market, it is poised to make a meaningful contribution to the product ecosystem by simplifying the search for accommodations, fostering community connections, and setting new standards for transparency and innovation in the Resell-Hub space. Welcome to the future of second-hand products and services solutions—welcome to Resell-Hub!

REFERENCEs

14. REFERENCES:

- www.google.com
- www.youtube.com/
- www.github.com/
- https://chat.openai.com/
- https://stackoverflow.com/
- https://codepen.io/
- Old Projects