**VIMAL TORMAL PODDAR BCA COLLEGE**

**VEER NARMAD SOUTH GUJARAT UNIVERSITY**

**PROJECT DOCUMENTATION**

**ON**

**E-COMMERCE**

**C2C PLATFORM**

**As Partial Requirement for the Degree**

**Of**

**Bachelor of Computer Applications**

**[B.C.A] [SEM-VI]**

**Year: 21-22**

Guided By:

# Asst Prof Mayur Maroliya

Submitted By:

1. Beauty Thakur Anilkumar Seat no. 2019024767
2. Jashoda Kalal Sohanlal Seat no. 2019024774

**ACKNOWLEDGMENT**

It gives me great pleasure in submitting this project entitled **“E-Commerce C to C Platform”** as a part of the curriculum of BCA (Semester VI).

We avail this opportunity to express my heartfelt gratitude to a number of people who extend their full support and co-operation in developing this project and also imparting Knowledge to me in various other domain of software technology.

We would like to take opportunity to thank my collage**, VIMAL TORMAL**

**PODDAR BCA COLLEGE, Surat.** for giving me this tremendous opportunity to work in the real-time project.

We heartily thank my project guide, **Asst Prof Mayur Maroliya** who was always there to guide me through the development of the project. He is one of the major sources behind the success of the project. I immensely appreciate the tips he has constantly given us during the project. It was an enormous pleasure to work with him.

We am thankful to thefaculty of the institute for their constant guidance not only during training period but also throughout college career.

Finally we would like to thank **My Parents** for their support throughout the project. I owe a special debt to our family & friends for their support blessing and encouragement for me.

**- Beauty Thakur**

**- Jashoda Kalal**

**INDEX**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** |  | **Contents** | | | | |  | **Page. No** | |
| **1.** |  | | **Introduction** | | | | | | |
|  |  | | **1.1** |  | Project Profile | |  | **4** | |
| **2.** |  | | **System Analysis** | | | | | | |
|  |  | | **2.1** |  | Hardware and Software Requirement | |  | **5** | |
|  |  | | **2.2** |  | System Overview | |  | **11** | |
|  |  | | **2.3** |  | Proposed System | |  | **12** | |
| **3.** |  | | **System Planning** | | | | | | |
|  |  | | **3.1** |  | | Feasibility Study |  | **13** | |
|  |  | | **3.2** |  | | Requirement Analysis and Data Gathering |  | **14** | |
|  |  | | **3.3** |  | | Process Model |  | **16** | |
|  |  | | **3.4** |  | | Task Dependency |  | **17** | |
| **4.** |  | | **System Design** | | | | | | |
|  |  | | **4.1** |  | Time line chart | |  | **20** | |
|  |  | | **4.2** |  | Use Case Diagram | |  | **22** | |
|  |  | | **4.3** |  | Data Dictionary | |  | **29** | |
|  |  | |  |  |  | |  |  | |
| **5.** |  | | **Screen Layouts** | | | |  | **36** | |
| **6.** |  | | **Testing** | | | | | | |
|  |  | | **6.1** |  | | Generating Testcases |  | | **56** |
|  |  | | **6.2** |  | | Unit Testing |  | | **57** |
|  |  | | **6.3** |  | | Integration Testing |  | | **57** |
|  |  | | **6.4** |  | | System Testing |  | | **59** |
|  |  | | **6.5** |  | | Alpha & Beta Testing |  | | **59** |
| **7.** |  | | **Future Enhancement** | | | |  | | **61** |
| **8.** |  | | **Bibliography** | | | |  | | **62** |

**1.Project Profile**

|  |  |
| --- | --- |
| **Project Title** | **E-Commerce C To C Platform** |
| **Project Definition** | ‘**E-Commerce** ’ Website basically contains:   1. Admin Side    * View Dashboard    * Manage User    * Manage Product Listing    * Manage Inquiry    * Manage Order    * Login/Logout 2. User Side    * Login/Logout    * Register User    * View Dashboard    * Add Product Listing    * Manage Account |
| **Duration** | 3 Months |
| **Project External Guide** | Asst Prof Mayur Maroliya |
| **Front End** | Html , Css ,JavaScript |
| **Back End** | Django ,Mysql |
| **Operating System** | Windows 10 [64 bits] |
| **Submitted By** | Jashoda Kalal (Seatno.:2019024767)  Beauty Thakur (Seatno.: 2019024774) |

**2. System Analysis**

2.1 **Hardware and Software Requirements**

* **Hardware Details**
* **Processor** : Intel Core i5
* **Ram** : 4GB (Minimum)
* **HDD** : 80GB (Minimum)
* **Moniter** : TFT(Thin Film Transistor)
* **Keyboard** : (85+keys)
* **Mouse** : Scroll Mouse(If Needed)

* **Software Details**

* **Software Requirements (for Development)**
* **Google Chrome**
* **Visual Studio Code**
* **Notepad++ editor**

* **Operation System (for development)**
* **Microsoft Windows 10**
* **For Client Side**
* Website

## Front-End:-

#### ➢ What is HTML

HTML is a language for describing web pages.

* HTML stands for **H**yper **T**ext **M**arkup **L**anguage
* HTML is a **markup** language
* A markup language is a set of markup **tags**
* The tags **describe** document content
* HTML documents contain HTML **tags** and plain **text**
* HTML documents are also called **web pages**

Python is a server side programming language so you would use it to fetch or update data on your server or you can use it to process some data then return the result. In other words it makes your website dynamic unlike HTML (semantics) which is static and CSS (design), Python allows you to show different and personalized information to your users.

####  What is CSS

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* Styles were added to HTML 4.0 **to solve a problem**
* **External Style Sheets** can save a lot of work
* External Style Sheets are stored in **CSS files**

**Three Ways to Insert CSS:**

* 1. Internal CSS
  2. External CSS
  3. Inline CSS

**JavaScript:**

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.).  There are also more advanced server side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than downloading files (such as realtime collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

### Example

document.getElementById("demo").innerHTML = "Hello JavaScript";

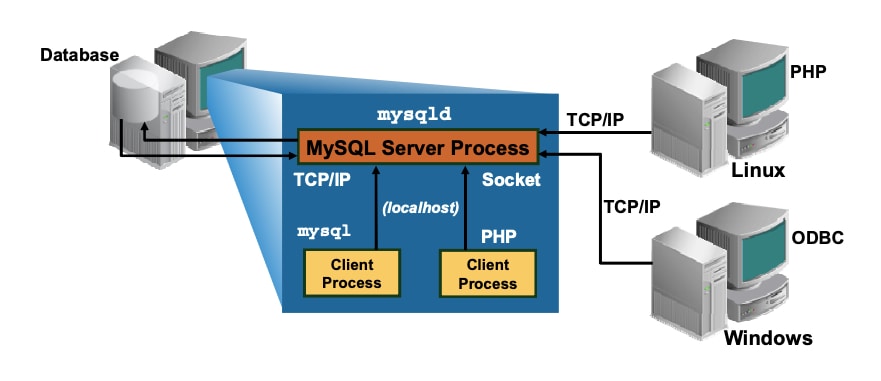
## Front-End:-

**MYSQL:**

In the era of rapid prototyping, we can get bright ideas, but sometimes they are not applicable if they take too much work. Often, the back-end is the limiting factor - many considerations never apply to server-side coding due to lack of knowledge or time.

Firebase is a Backend-as-a-Service(BaaS) which started as a YC11 startup. It grew up into a next-generation app-development platform on Google Cloud Platform. Firebase (a NoSQLjSON database) is a real-time database that allows storing a list of objects in the form of a tree. We can synchronize data between different devices.

**MYSQL DATABASE SERVER**



❖ **Data Types:**

* Many data types: signed/unsigned integers 1,2,3,4, and 8 byteslong,FLOAT,DOUBLE,CHAR,VARCHAR,TEXT,BLOB,DATE,TIME,DAT ETIME,TIMESTAMP, YEAR, SET, ENUM, and Open GIS special types.
* Fixed-length and variable-length records.

➲ **Security:**

* Privilege and password systems that is very flexible and secure, and that allows hostbased verification .Password are secure because traffic is encrypted when you connect to a server.

# Django:

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!

### Installing an official release with pip

This is the recommended way to install Django.

1. Install [pip](https://pip.pypa.io/). The easiest is to use the [standalone pip installer](https://pip.pypa.io/en/latest/installation/). If your distribution already has **pip** installed, you might need to update it if it’s outdated. If it’s outdated, you’ll know because installation won’t work.
2. Take a look at [venv](https://docs.python.org/3/tutorial/venv.html). This tool provides isolated Python environments, which are more practical than installing packages systemwide. It also allows installing packages without administrator privileges. The [contributing tutorial](https://docs.djangoproject.com/en/4.0/intro/contributing/) walks through how to create a virtual environment.
3. After you’ve created and activated a virtual environment, enter the command:

**$** python -m pip install Django

### installing the development version

1. Make sure that you have [Git](https://git-scm.com/) installed and that you can run its commands from a shell. (Enter **git help** at a shell prompt to test this.)
2. Check out Django’s main development branch like so:

**$** git clone https://github.com/django/django.git

This will create a directory **django** in your current directory.

1. Make sure that the Python interpreter can load Django’s code. The most convenient way to do this is to use a virtual environment and [pip](https://pip.pypa.io/). The [contributing tutorial](https://docs.djangoproject.com/en/4.0/intro/contributing/) walks through how to create a virtual environment.
2. After setting up and activating the virtual environment, run the following command:

**$** python -m pip install -e django/

This will make Django’s code importable, and will also make the **django-admin** utility command available. In other words, you’re all set!

When you want to update your copy of the Django source code, run the command **git pull** from within the **django** directory. When you do this, Git will download any changes.

**Postgres.app**

Postgres.app is a full-featured PostgreSQL installation packaged as a standard Mac app. It includes everything you need to get started, and we’ve even included the popular extension [PostGIS](http://postgis.net/) for geo data.

Postgres.app has a beautiful user interface and a convenient menu bar item. You never need to touch the command line to use it – but of course we do include all the necessary [command line tools](https://postgresapp.com/documentation/cli-tools.html) and header files for advanced users.

Postgres.app can install minor updates automatically, so you get bugfixes as soon as possible.

**Installing Postgres.app**

* Download   ➜   Move to Applications folder   ➜   Double Click

If you don't move Postgres.app to the Applications folder, you will see a warning about an unidentified developer and won't be able to open it.

* Click "Initialize" to create a new server
* Configure your $PATH to use the included command line tools (optional):

sudo mkdir -p /etc/paths.d &&  
echo /Applications/Postgres.app/Contents/Versions/latest/bin | sudo tee /etc/paths.d/postgresapp

**2.2** **System Overview**

1. The E-Commerce C to C platform project is online shoping based on an idea of the buy and sell online product.
2. Electronic commerce, commonly written as E-Commerce, is the trading in products or services using computer networks, such as the Internet.
3. Providing or participating in online marketplaces, which process third-party business- to-consumer or consumer-to-consumer sales.

**Module:-**

* **User Registration:** Coustomer Can Register in E-commerce website .
* **User Login:** Coustomer can login to system and jump to the home screen.
* **Product Gallery:** Coustomer can view the various Counsumer post.
* **Add Post:** Counsumer can buy the product.
* **User Profile:** Coustomer can view their profile and also edit their profile
* **Purchase product :** Coustomer Buy the product.

2.3 **Purposed System**

**2.2.1** **Scope**

* C2C websites and similar platforms make money from fees charged to sellers for listing items for sale, adding on promotional features, and facilitating credit card transactions.
* E-Commerce is buying and selling, marketing, servicing delivery and payment of products, service and information over internet, intranets, extranets and other networks, between an inter-networked enterprise and its prospects, customers suppliers and other business partners.
* Online selling and purchasing offer innumerable benefits to both sellers and buyers, and these advantages are also the reasons for the rising scope of eCommerce.

**2.2.2** **System Objective (Goals)**

* Consumer to consumer, or C2C, is the business model that facilitates commerce between private individuals. Whether it's for goods or services, this category of [e-commerce](https://www.businessnewsdaily.com/4872-what-is-e-commerce.html) connects people to do business with one another.
* E-commerce is the route for purchasing goods and services online. The money transactions are done through online thus leads to the digital economy.
* The C2C is to enable these relationships, helping buyers and sellers locate each other. Customers can benefit from the competition for products and easily find products that may otherwise be difficult to locate.

**3. System Planning**

**3.1 Feasibility Study:**

* A Feasibility Study plays very important role in the development of any system. But when it is the case of development of any software then it‘s important to increases much more because in this case one should be very clear about the availability of the time and resources.
* Before starting the development of the software one should give considerable amount of time for feasibility study because the successful completion of the Project depends upon its Feasibility.
* **Operating Feasibility:**

During the operational feasibility one has to take care about the system‘s impact on the other systems and its tendency to fulfil the requirements of the organization while working interfering to the fulfil the requirements while working without interfering to the other existing system.

* **Technical Feasibility:**

The Technical Feasibility involves the study about the availability of the tools and hardware as well as software requirements for the development and the implementation of the system. During the study it was decided to use the Python and Django, Mysql.

* **Resources:**

When developing any system, we need proper environment resources as well as man power Also our team members are acquainted to such type of development environment. For gathering more help when development application we also provision of an internet connection.

* **Time Duration:**

We have one month for developing entire system. It is less sufficient time for developing such relatively more complex application. We prepared time schedule which can efficient manage proper time required for project work.

**3.2 Requirement Analysis Data Gathering:**

To develop any software system, it is most important to identify the user requirement in very same manner. Also to function properly, all interfaces of proposed system with bounding system must be identified. The correct system is that that satisfied user requirements. Therefore it very important to analyse the existing system and to document the software requirement specification for the proposed system, which in turn provides the base for development of the proposed system.

Along with our technical configuration, we simultaneously started the system study and analysis. During this method, initially we started with studying the system specification document and understand the system and unveil the base system involved.

Our project guide held a series of lecture to give the required knowledge about the system. During the lectures, we also had the question & answer session at the end, all that question to have a clear about the expected system.

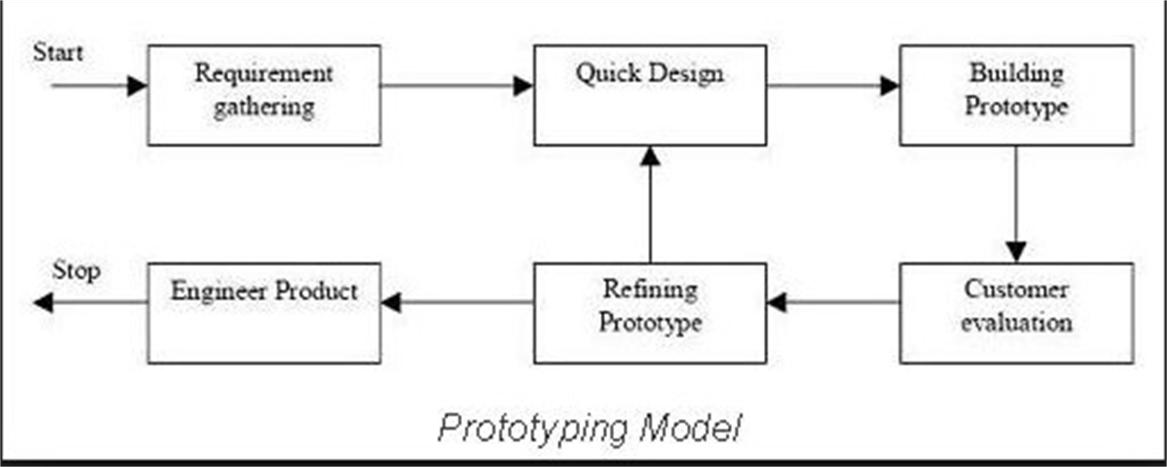
* + - Information gathering in any organization is not easy task.
    - It has to gathered to be organized way so that
      1. No system details are left out.
      2. Right problems to be identified.
      3. Repetitive work is avoided.
      4. Wrong or incomplete information are not to be collected.
    - To do this proper search strategy must be decided first, search strategy includes modelling methods to make sense out of information so collected.
    - Here an overall idea about the search methods or fact gathering techniques which are used while gathering information, they are
      1. Interviewing
      2. Record inspection
      3. Observation

1. **Interviews:**
   * + Interview allows analyst to collect or gather the information from the individual or group who are generally the current user of the existing system.
     + As far as interview is concern it is a time consuming process
     + It is a basic source of qualitative information
     + It allows the analyst to discover areas if misunderstanding. Indication of resistance of existing system.

1. **Record Inspection**
   * + It is said to better believe in records than people.
     + Thus a good analyst always gets facts from document.
     + An existing system can be better understood by the existing documents, forms and files.
     + In record review analyst examine information that has been recorded about the system and users.
     + Records inspection can be done at the beginning of study as introduction or later as a basic for comparing actual operation with what the records indicate should be happening.
     + Records may include
       - 1. Written policy manuals.
         2. Rules and regulation.
         3. Standard operating procedure used in the organization.
         4. Forms and documents.
2. **Observation**
   * Observation can bring in missed facts, new ways to improve the existing procedures, duplication work done inadvertently, etc
   * Observation can bring in what other fact-finding methods can‘t! But this task is delicate because people do not like to observe when they work.
   * It is not the quantity of time observed is important but the unorthodox angles of observation of work content and methods are going to be rewarding.
   * Observation gives analyst the opportunity to go behind the scenes in an organization to learn inside story to discover how things work in new areas of information.
   * Observation can look for:
     + 1. Operational inefficiencies.
       2. Alternate routes and procedures.
       3. Interruptions in the normal flow of work,
       4. The usage of files and documents.

* On site observation provides close view of the working of the real system.
* He can observe people, objects, documents and occurrences of events.
  1. **Process Model:**

**PROTOTYPE MODEL**



The incremental approach attempts to combine the waterfall sequence with some of the advantages of prototyping. This approach is favoured by many object-oriented practitioners. It basically divides the overall project into a number of increments. Then it applies the waterfall model to each increment. The system is put into production when the first increment is delivered. As time passes additional increments are completed and added to the working system. This approach is favoured by many object-oriented practitioners.

**Incremental Phases :-**

### Inception :-

During the inception phase, the purpose, business rationale, and scope of the project is determined. This is similar to the feasibility analysis that is done in other life cycles.

### Construction :-

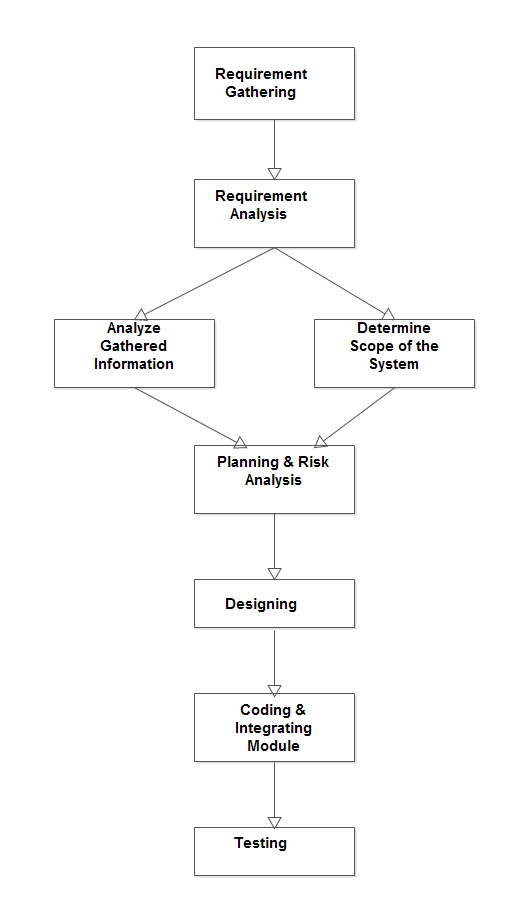
The construction phase builds increments of the system. Each increment is developed using a waterfall approach. This includes detailed analysis and design for the use cases in the increment and coding and testing of the event processors that implement the sequence of events defined by the use cases. The result is production quality software that satisfies a subset of the requirements and is delivered to the end users. Work on different increments may be done in parallel.

### Transition :-

The transition phase (not shown in the figure) is the last phase in the project. This may includesuch things a performance tuning and rollout to all users.

**3.4. Task Dependency**

* **Tasks List :**
  1. **I**nitial Phase
     1. Requirement Gathering
     2. Feasibility Study
  2. Requirement Analysis
     1. Identify Program Flow
     2. Identify Constraints
     3. Finalize Requirements
  3. Project Scheduling
  4. Project Design
     1. Conceptual Diagram
     2. Identify flow of project
     3. Identify page layouts
  5. Coding
     1. Develop **Admin site** = Admin can manage the user.
     2. Develop **User site** = Usercan add and view the product .
     3. Createcss and implement it.
  6. Testing
     1. Testing pages individually
     2. Error correction
  7. Releasing
* **Task Dependency Diagram**



## 4. System Design

**4.1 Time Line Chart:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Month1** | | | | **Month** | | **2** | | **Month3** | | | |
| **1) Requirement analysis** |  |  |  |  |  |  |  |  |  |  |  |  |
| Collected requirement from our guide |  |  |  |  |  |  |  |  |  |  |  |  |
| Analyze gathered information |  |  |  |  |  |  |  |  |  |  |  |  |
| Determine scope of the system |  |  |  |  |  |  |  |  |  |  |  |  |
| **Milestone: req. Analysis complete** |  |  |  |  |  |  |  |  |  |  |  |  |
| **2) Planning and risk analysis** |  |  |  |  |  |  |  |  |  |  |  |  |
| Analyze data for possible risks |  |  |  |  |  |  |  |  |  |  |  |  |
| Identify technical risks |  |  |  |  |  |  |  |  |  |  |  |  |
| Determine different modules |  |  |  |  |  |  |  |  |  |  |  |  |
| **Milestone: Planning completed** |  |  |  |  |  |  |  |  |  |  |  |  |
| **3) Designing** |  |  |  |  |  |  |  |  |  |  |  |  |
| Design basic interface of the site |  |  |  |  |  |  |  |  |  |  |  |  |
| Design database tables |  |  |  |  |  |  |  |  |  |  |  |  |
| Design web form for modules |  |  |  |  |  |  |  |  |  |  |  |  |
| **Milestone: designing completed** |  |  |  |  |  |  |  |  |  |  |  |  |
| **4) Coding and integrating modules** |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement logic for different modules |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement database connectivity |  |  |  |  |  |  |  |  |  |  |  |  |
| Integrate different modules |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement Report |  |  |  |  |  |  |  |  |  |  |  |  |
| **Milestone: coding completed** |  |  |  |  |  |  |  |  |  |  |  |  |
| **5) Testing** |  |  |  |  |  |  |  |  |  |  |  |  |

#### UML diagram

* **UML is a language for specifying, constructing and documenting software system**
  + General Purpose Modeling Language
  + Merges Modeling Element from Run Baugh ,Jacobson and others
  + Object Oriented Analysis and Design
  + Graphical Notation supporting numerous diagrams
  + Extensible notation (stereotypes)
  + Extensible Semantics(Object constraints Language)

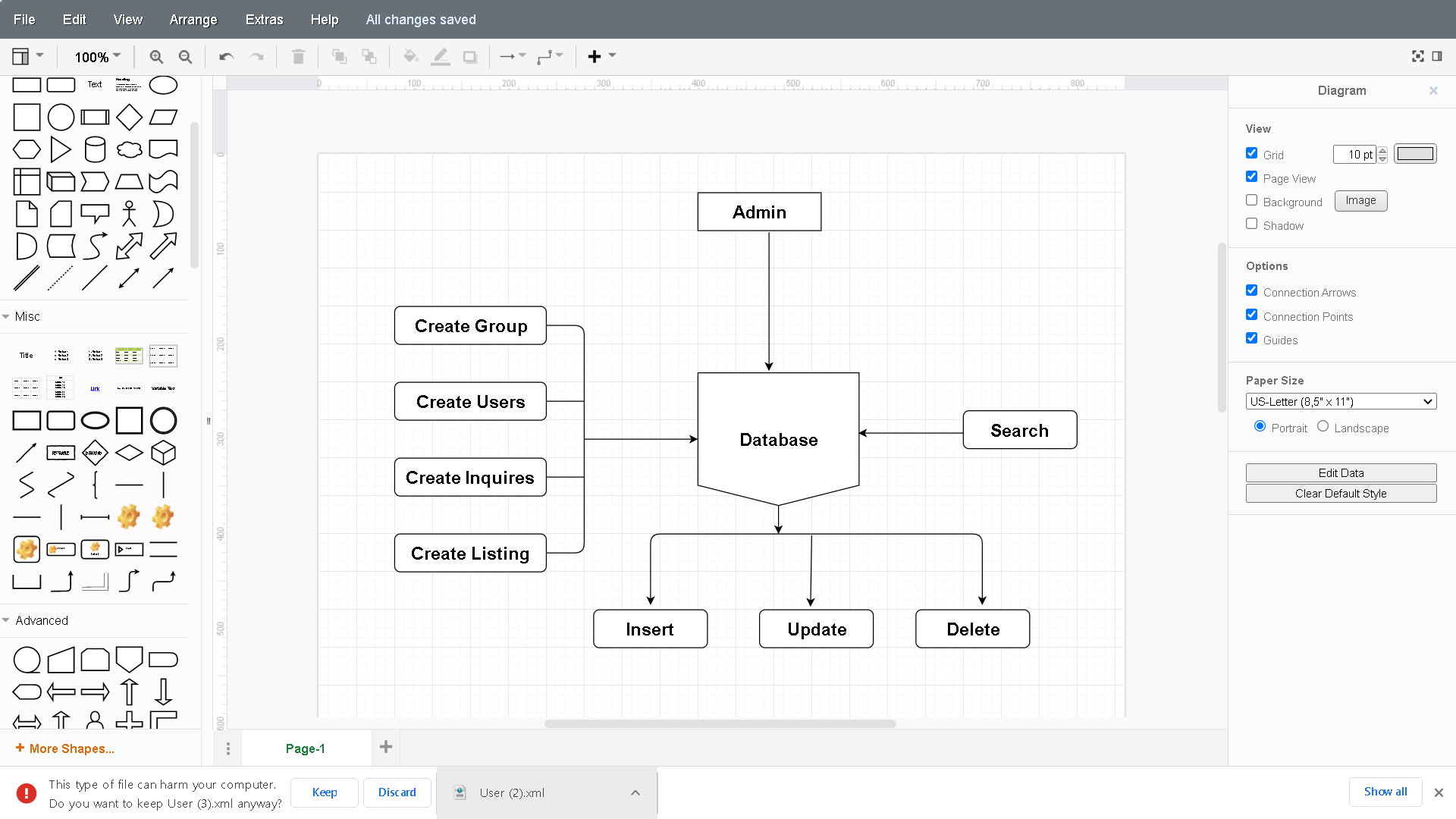
#### Components

* + Classes, Objects, and Packages
  + States and Activities
  + Actors and Use-Cases

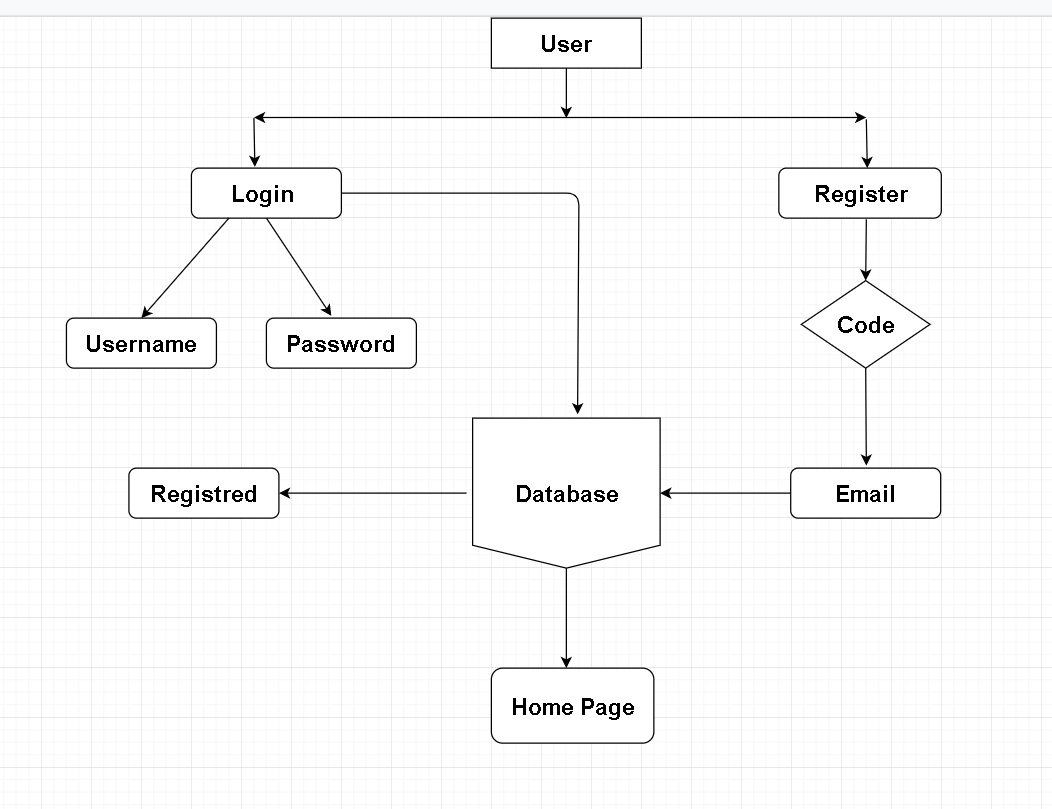
#### Connectors

* + Basically three types :- Communication (control and data), Containment
  + Restricted in what types of components they link
  + First class citizens
  1. **Use Case Diagram**

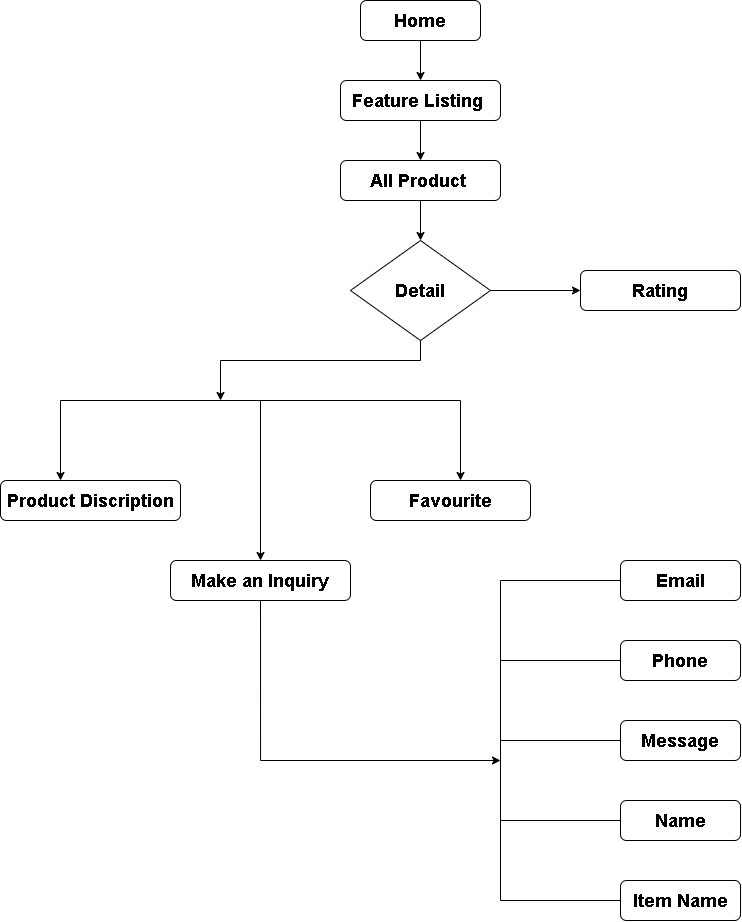
**Admin Side**



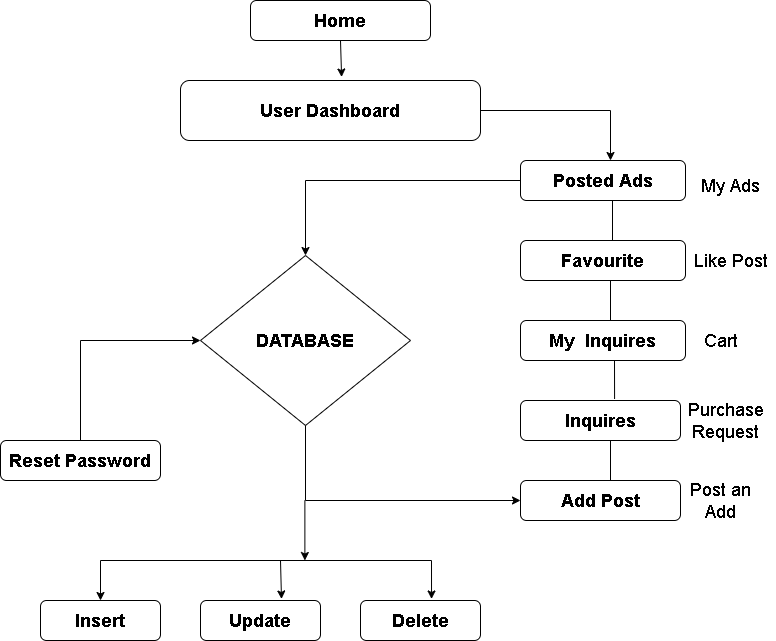
**User Side**



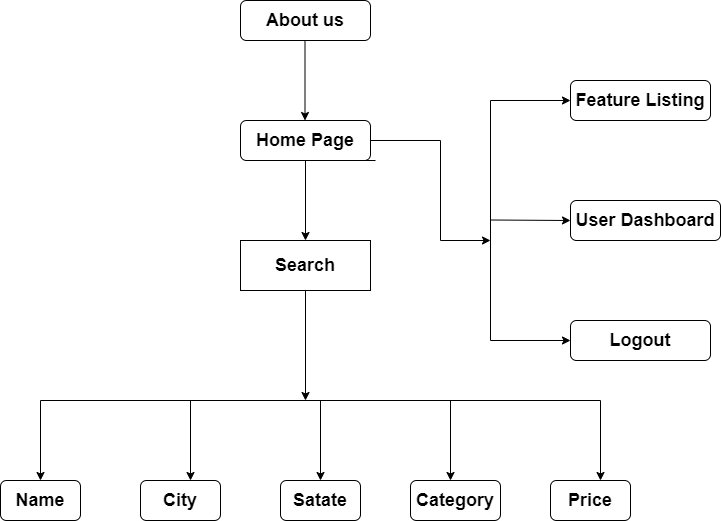
**Home Page**



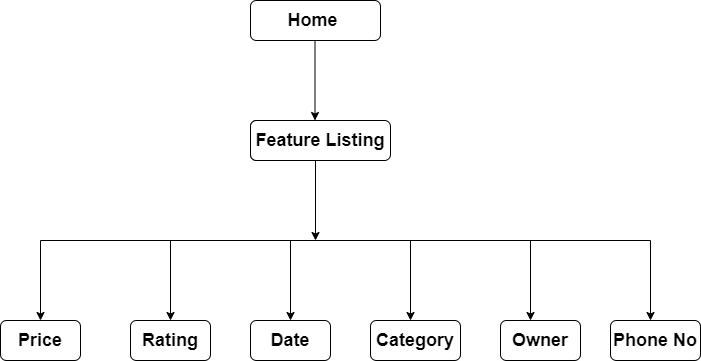
**User Dashboard**



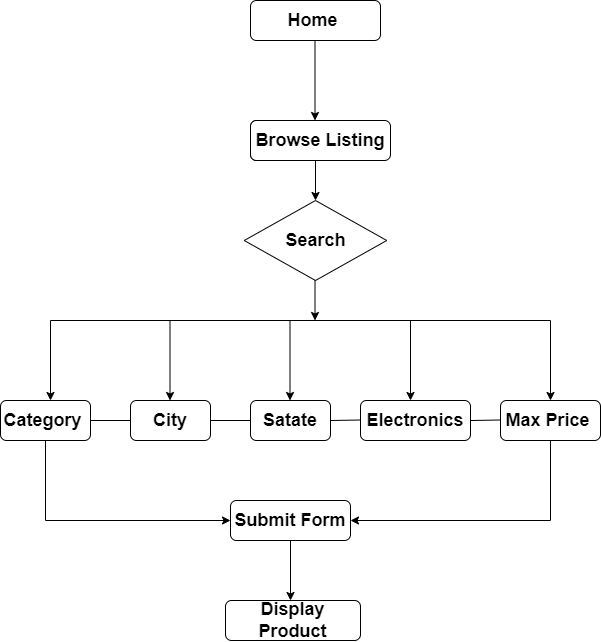
**About us**



**Added Post Detail**



**Search Product**



* 1. **Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Srno** |  | **Table Name** |  |  |
|  | | | | |
| **1.** | **auth\_group** | | | |
|  | | | | |
| **2.** | **auth\_group\_permission** | | | |
|  | | | | |
| **3.** | **auth\_permission** | | | |
|  | | | | |
| **4.** | **core\_user** | | | |
|  | | | | |
| **5.** | **core\_user\_groups** | | | |
|  | | | | |
| **6.** | **core\_user\_users\_permission** | | | |
|  | | | | |
| **7.** | **django\_admin\_log** | | | |
|  | | | | |
| **8.** | **django\_content\_type** | | | |
|  | | | | |
| **9.** | **django\_migrations** | | | |
|  | | | | |
| **10.** | **django\_session** | | | |
|  | | | | |
| **11.** | **inquiry\_inquiry** | | | |
|  | | | | |
| **12.** | **listing\_listing** | | | |
|  | | | | |

* + 1. **Table Name :- auth\_group**

**Primary Key :-** id

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Column** | **Data Type** | **Null** |
| **1.** | Id | Int(11) | No |
| **2.** | name | varchar(150) | No |

* + 1. **Table Name :- auth\_group\_permissions**

**Primary Key :-** id

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data Type** | **Null** |
| **1.** | Id | Int(11) | No |
| **2.** | group\_id | Int(11) | No |
| **3.** | Permission\_id | Int(11) | No |

* + 1. **Table Name :- auth\_permissions**

**Primary Key :-** id

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Column** | **Data Type** | **Null** |
| **1.** | Id | int(11) | No |
| **2.** | name | varchar(255) | No |
| **3.** | content\_type\_id | int(11) | No |
| **4.** | codename | varachar(100) | No |

* + 1. **Table Name :- core\_user**

**Primary Key :-** id

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Column** | **Data Type** | **Null** |
| **1.** | id | int(11) | No |
| **2.** | password | varchar(128) | No |
| **3.** | last\_login | datetime(6) | Yes |
| **4.** | is\_superuser | tinyint(1) | No |
| **5.** | username | varchar(200) | No |
| **6.** | email | varchar(255) | No |
| **7.** | first\_name | varchar(100) | No |
| **8.** | last\_name | varchar(100) | No |
| **9.** | favourites | varchar(200) | No |
| **10.** | rate\_listing | varchar(200) | No |
| **11.** | phone | bigint(20) | No |
| **12.** | is\_active | tinyint(1) | No |
| **13.** | is\_staff | tinyint(1) | No |

* + 1. **Table Name :- core\_user\_groups**

**Primary Key :-** id

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Column** | **Data Type** | **Null** |
| **1.** | Id | int(11) | No |
| **2.** | user\_id | int(11) | No |
| **3.** | group\_id | int(11) | No |

* + 1. **Table Name :- core\_user\_users\_permissions**

**Primary Key :-** id

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data Type** | **Null** |
| **1.** | Id | int(11) | No |
| **2.** | user\_id | int(11) | No |
| **3.** | permission\_id | int(11) | No |

**7. Table Name :- django\_admin\_log**

**Primary Key :-**  id

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data type** | **Null** |
| **1.** | Id | Int(11) | No |
| **2.** | action\_name | datetime(6) | No |
| **3.** | object\_id | longtext | Yes |
| **4.** | object\_repr | varchar(200) | No |
| **5.** | action\_flag | smallint(5) | No |
| **6.** | change\_message | longtext | No |
| **7.** | content\_type\_id | int(11) | Yes |
| **8.** | userid | int(11) | No |

**8. Table Name :- django\_content\_type**

**Primary Key :-**  id

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Column** | **Data Type** | **Null** |
| **1.** | Id | int(11) | No |
| **2.** | app\_label | varchar(100) | No |
| **3.** | model | varchar(100) | No |

**9. Table Name :- django\_migrations**

**Primary Key :-**  id

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data Type** | **Null** |
| **1.** | id | Int(11) | No |
| **2.** | app | varchar(255) | No |
| **3.** | name | varchar(255) | No |
| **4.** | appiled | datatime(6) | No |

**10. Table Name :- django\_session**

**Primary Key :-**  session\_key

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data Type** | **Null** |
| **1.** | session\_key | varchar(40) |  |
| **2.** | session\_data | session\_data |  |
| **3.** | expiry\_date |  |  |

**11. Table Name :- inquiry\_ inquiry**

**Primary Key :-**  id

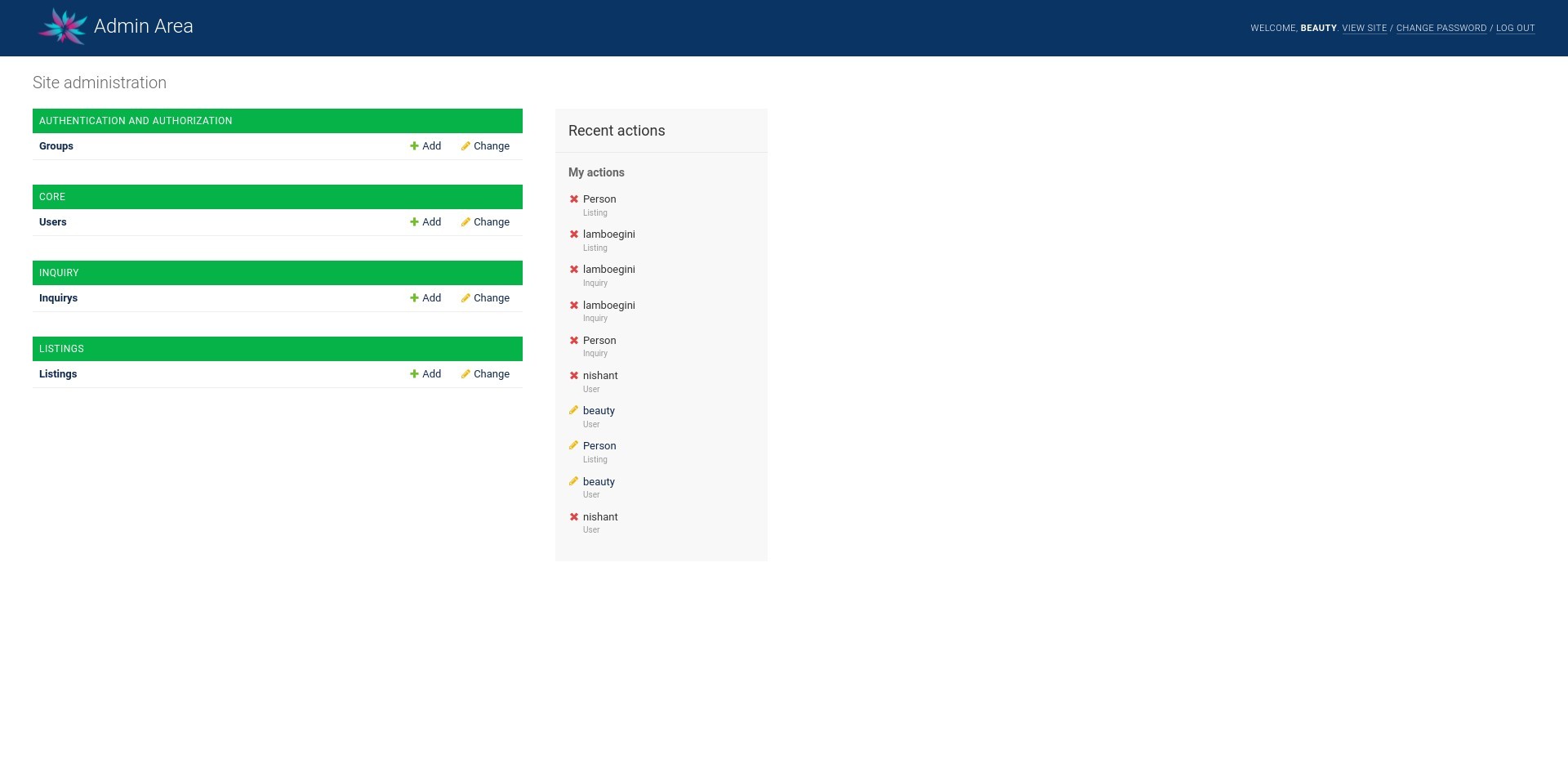
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data Type** | **Null** |
| **1.** | id | int(11) | No |
| **2.** | listing | varchar(200) | No |
| **3.** | listing\_id | int(11) | No |
| **4.** | name | varchar(100) | No |
| **5.** | email | varchar(255) | No |
| **6.** | phone | varchar(100) | No |
| **7.** | message | longtext | No |
| **8.** | contact\_date | date | No |
| **9.** | user\_id | int(11) | No |
| **10.** | owner\_id | int(11) | No |

**12. Table Name :- listing\_listing**

**Primary Key :-**  id

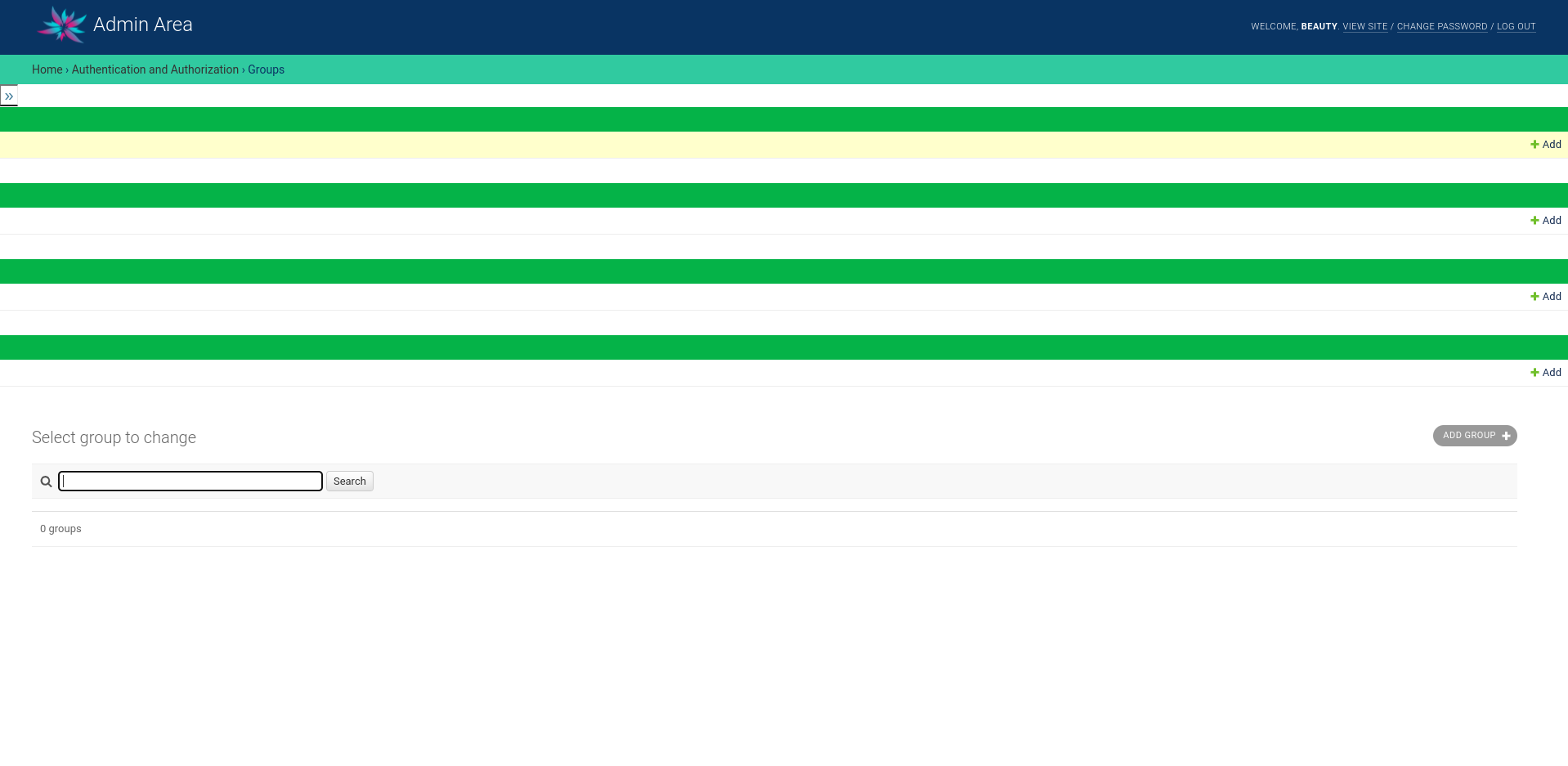
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Column** | **Data Type** | **Null** |
| **1.** | id | Int(11) | No |
| **2.** | title | varchar(100) | No |
| **3.** | category | varchar(100) | No |
| **4.** | address | varchar(100) | No |
| **5.** | city | varchar(100) | No |
| **6.** | state | varchar(100) | No |
| **7.** | zipcode | varchar(20) | No |
| **8.** | description | longtext | No |
| **9.** | price | Int(11) | No |
| **10.** | photo\_main | varchar(100) | No |
| **11.** | photo\_1 | varchar(100) | No |
| **12.** | photo\_2 | varchar(100) | No |
| **13.** | photo\_3 | varchar(100) | No |
| **14.** | photo\_4 | varchar(100) | No |
| **15.** | photo\_5 | varchar(100) | No |
| **16.** | photo\_6 | varchar(100) | No |
| **17.** | is\_published | tinyint(1) | No |
| **18.** | list\_date | date | No |
| **19.** | owner\_id | Int(11) | No |
| **20.** | no\_of\_rating | Int(11) | Yes |
| **21.** | total\_rating | Int(11) | Yes |

* + - 1. **Screen Layout**
* **Admin Home Page**



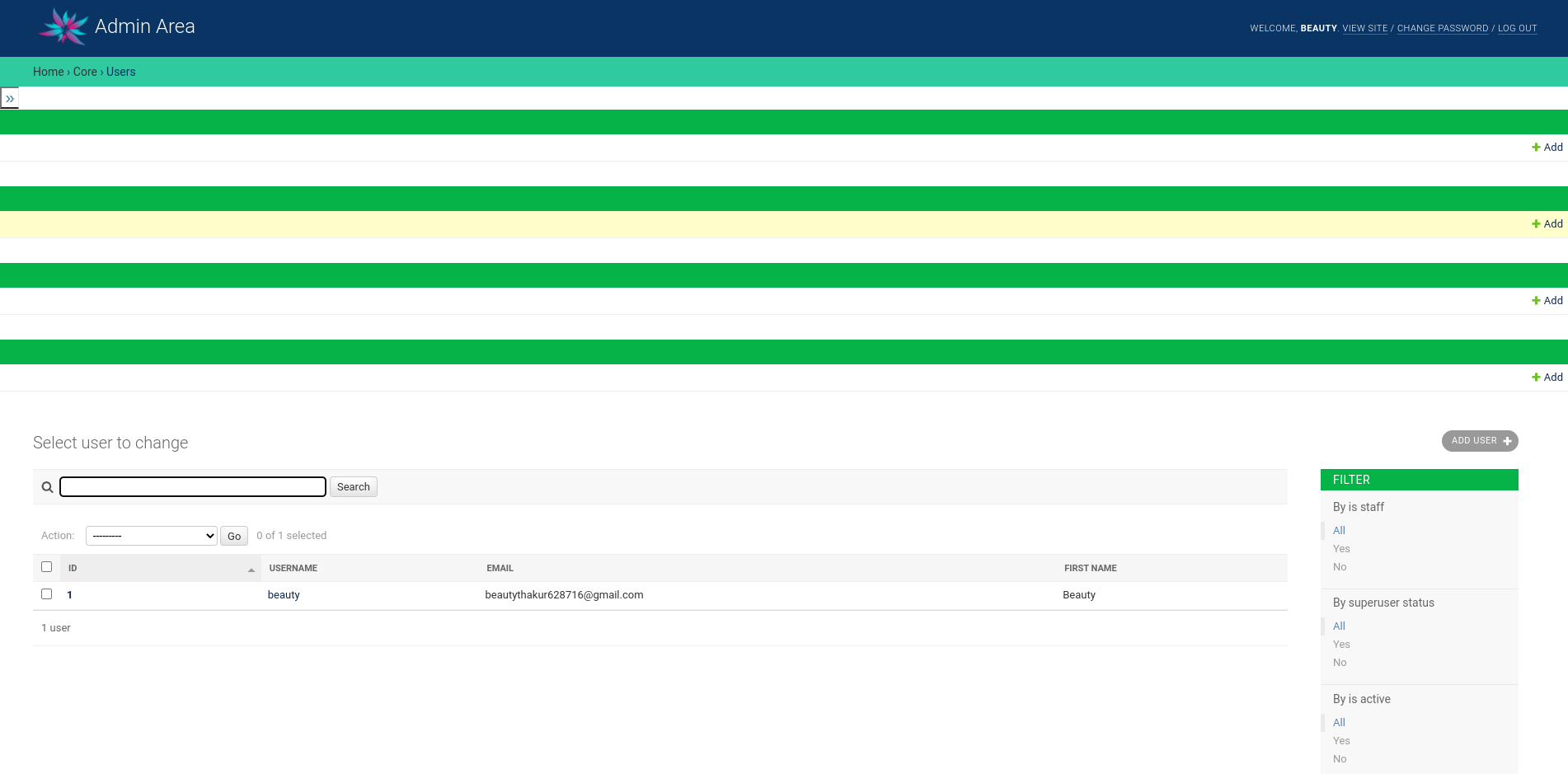
**Description : This page is for admin home page.**

* **Admin Group Page**



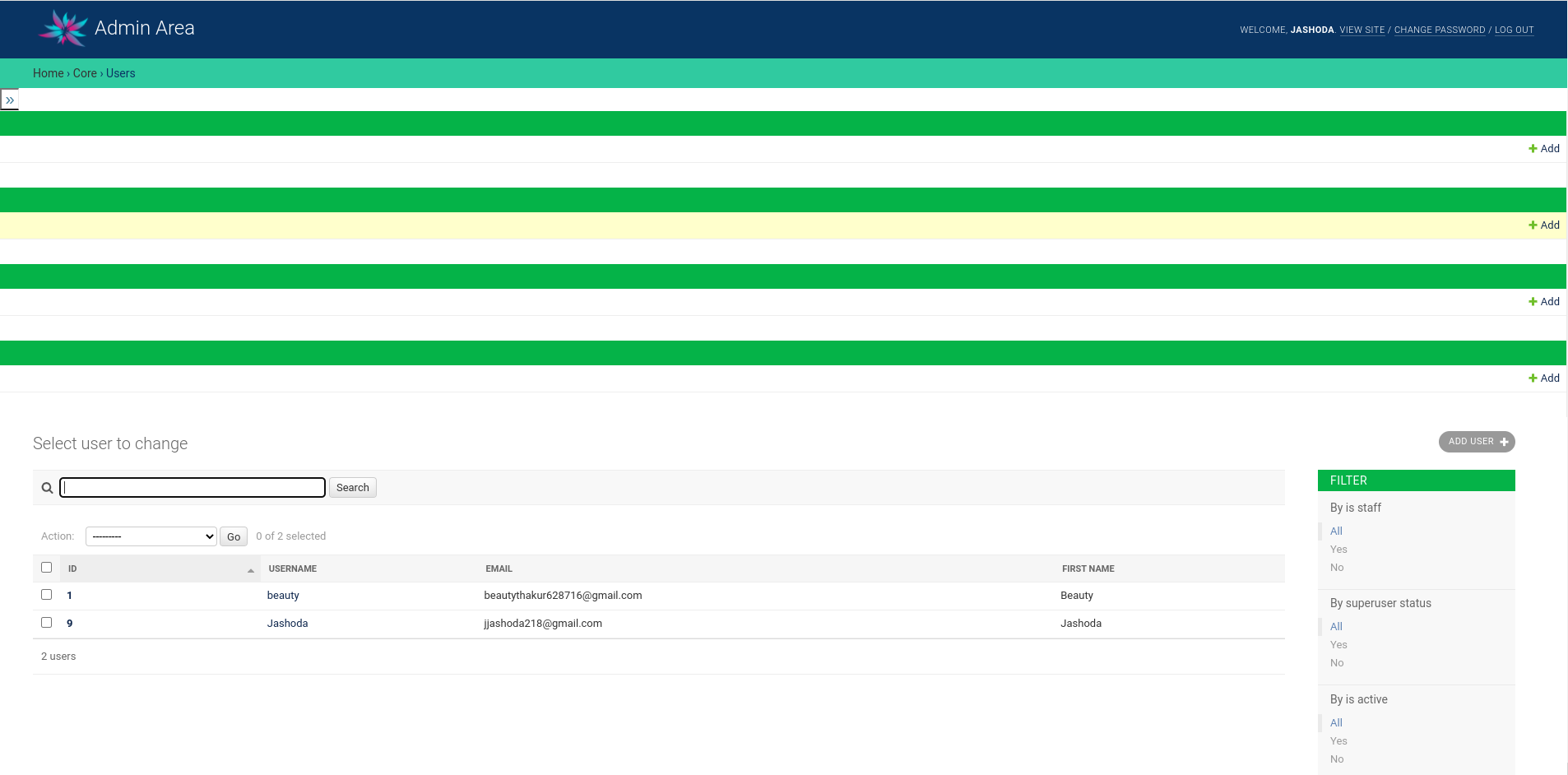
**Description : This page is for admin manage group page.**

* **Admin User Page**



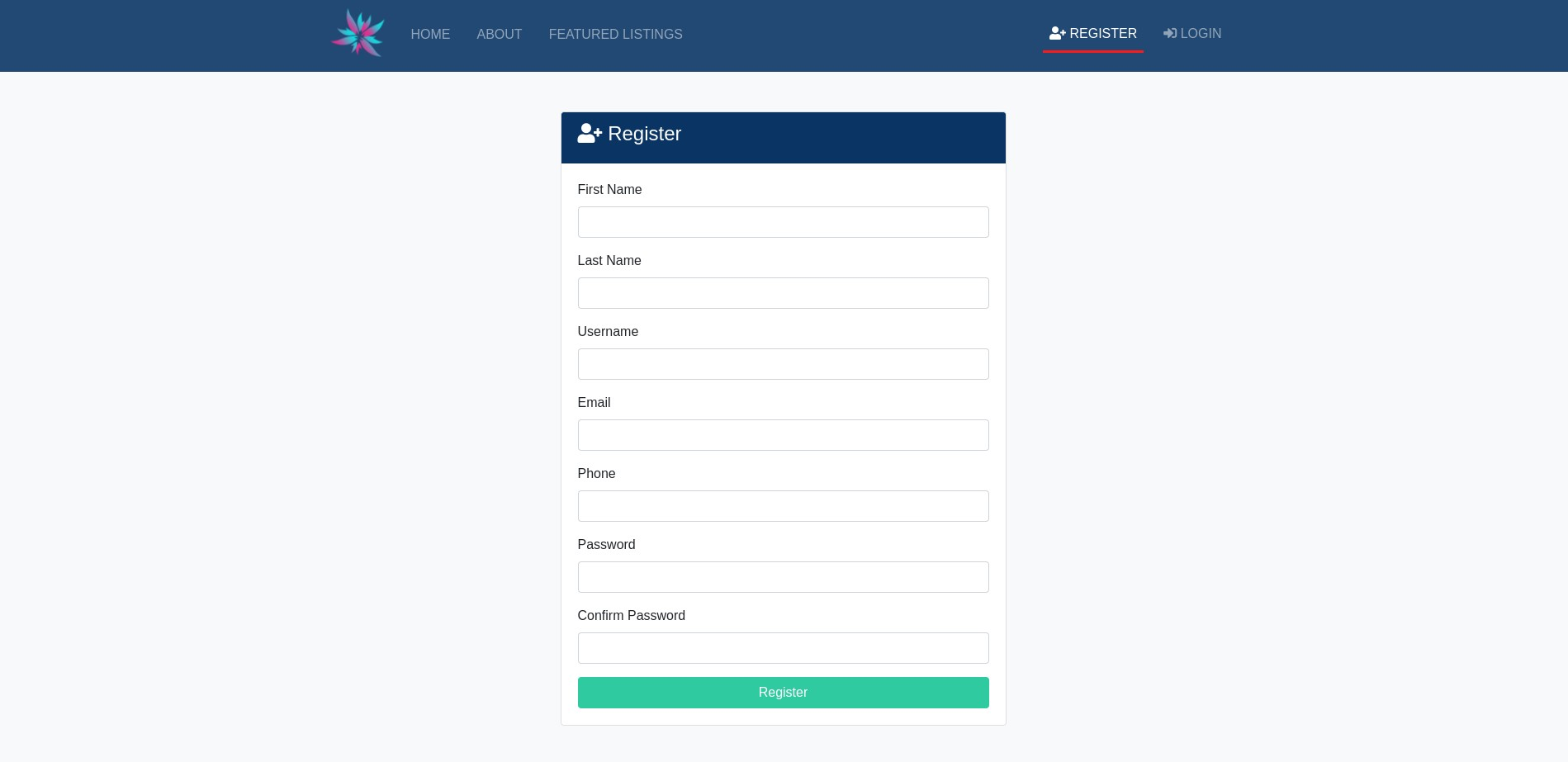
**Description : This page is for admin manage user page.**

* **Admin Listing Page**



**Description : This page is for admin manage Listing page.**

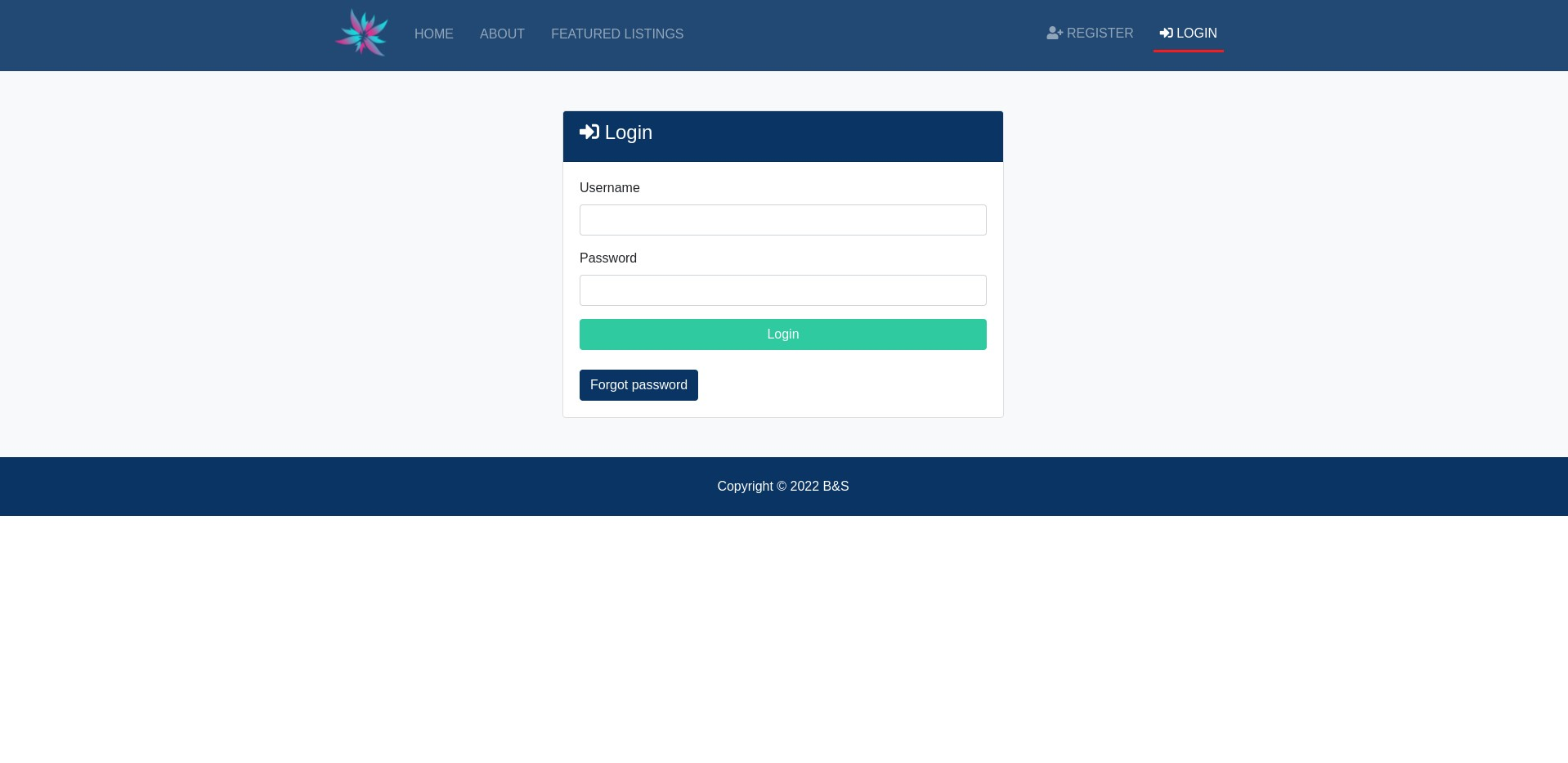
* **User Register Page**



**Description : This page is USER Register Page. Here Create**

**the new user profile.**

* **User Login Page**

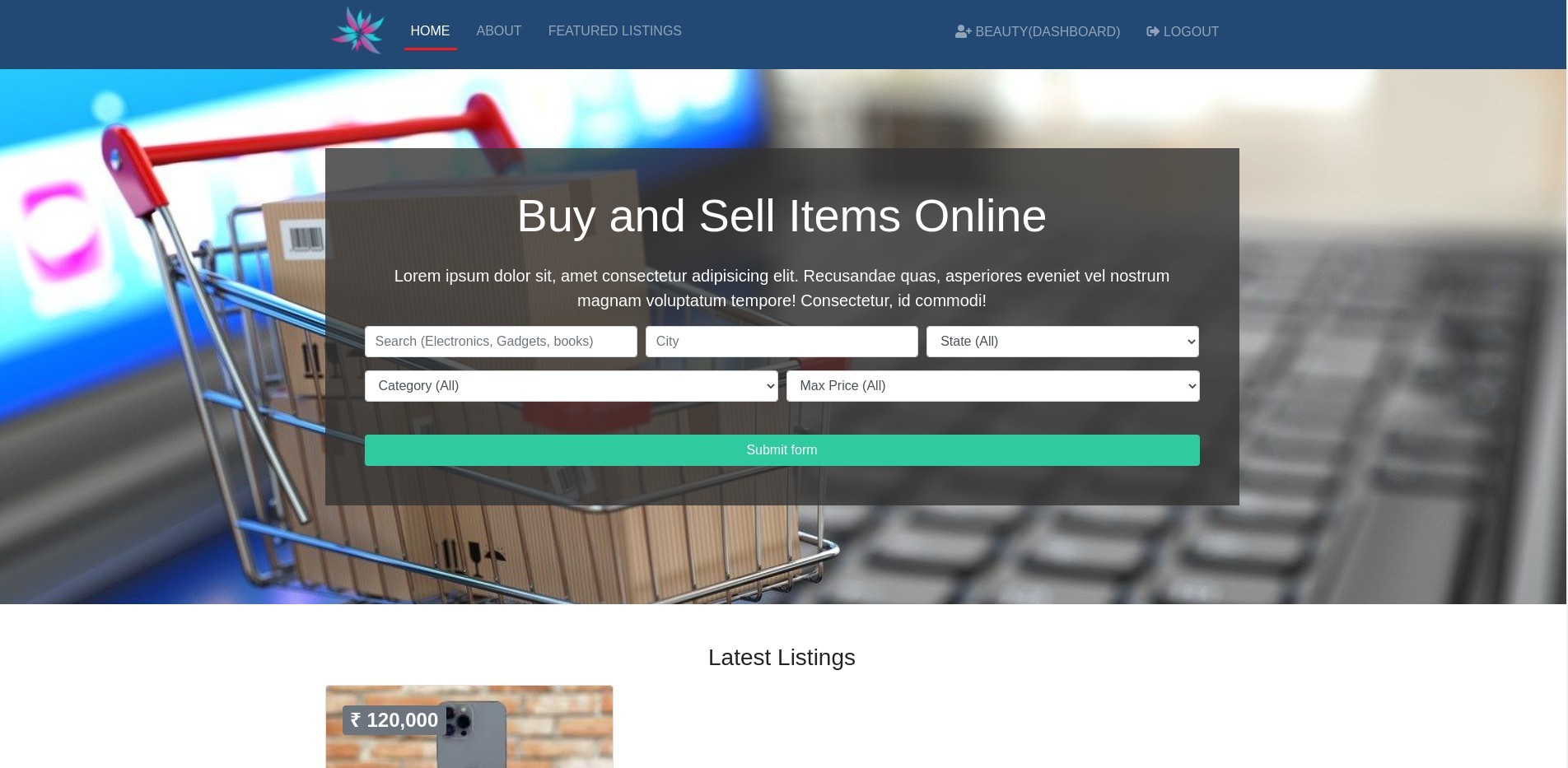


**Description : This page is USER login Page. Here user fill the**

**own username and password then switch to home**

**page.**

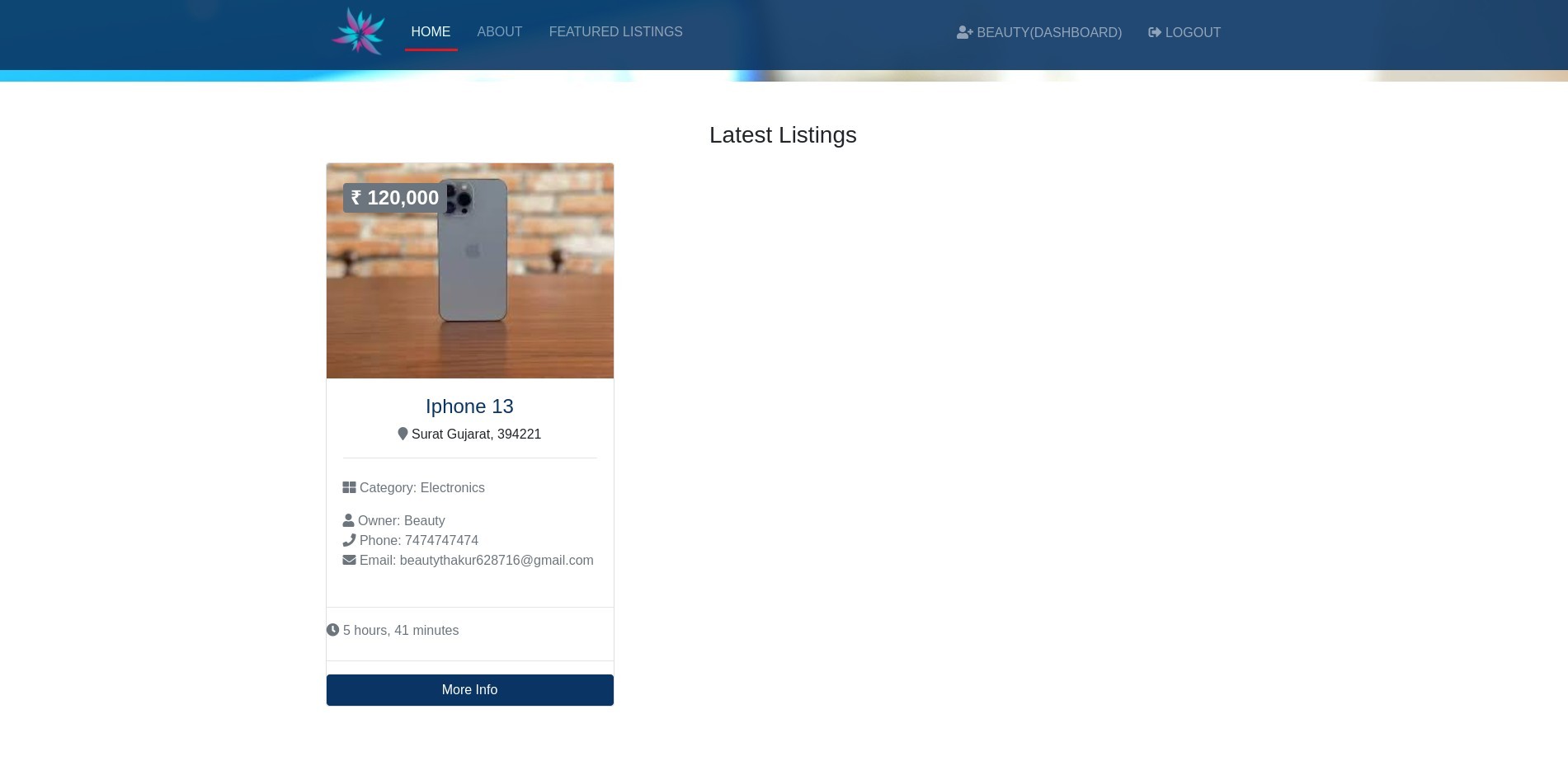
* **Home Page**



**Description : This page is home Page. Here costumer buy and**

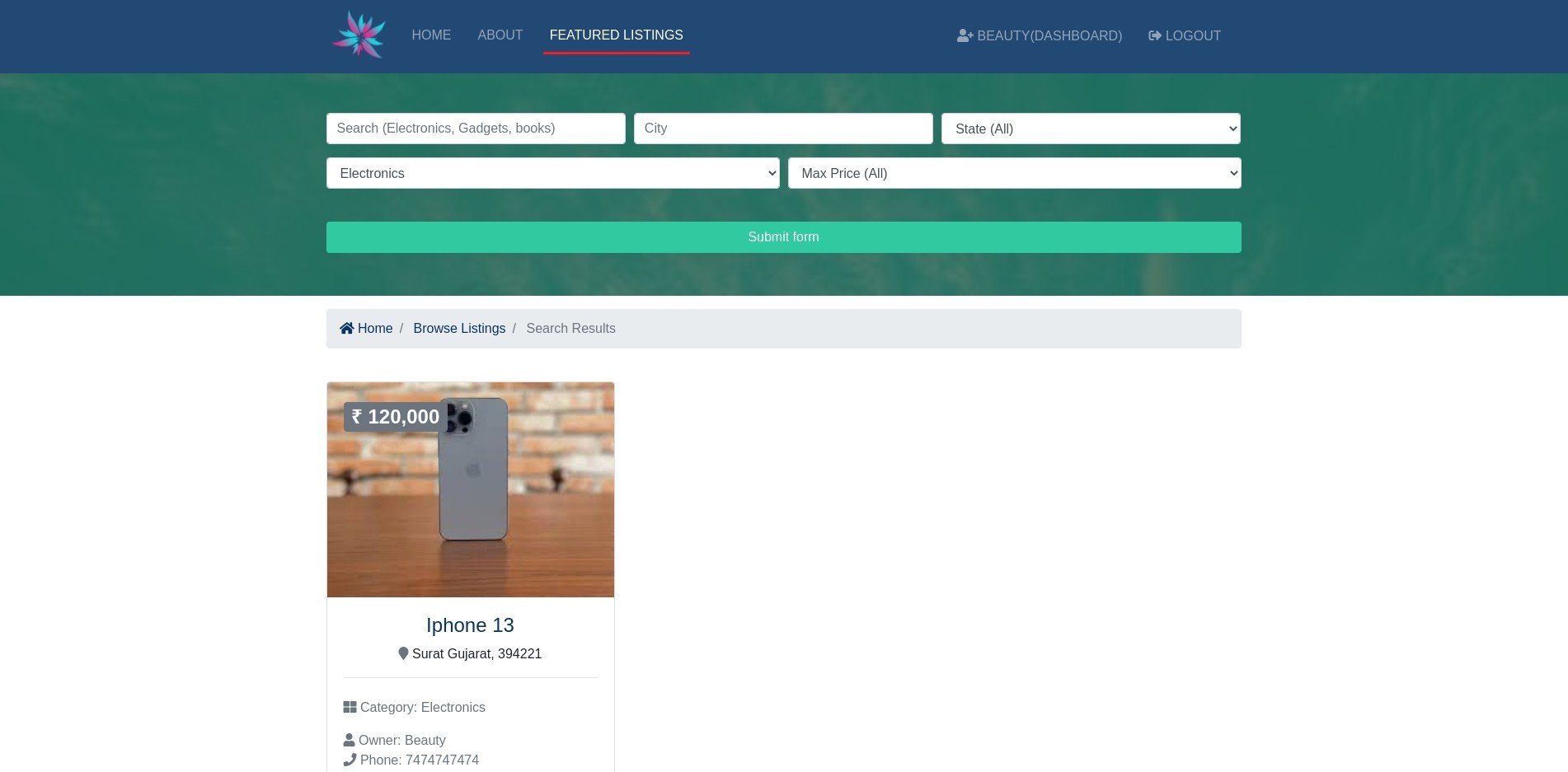
**sell the product.**

* **View Latest Product**



**Description : This page is View the Latest listing product.**

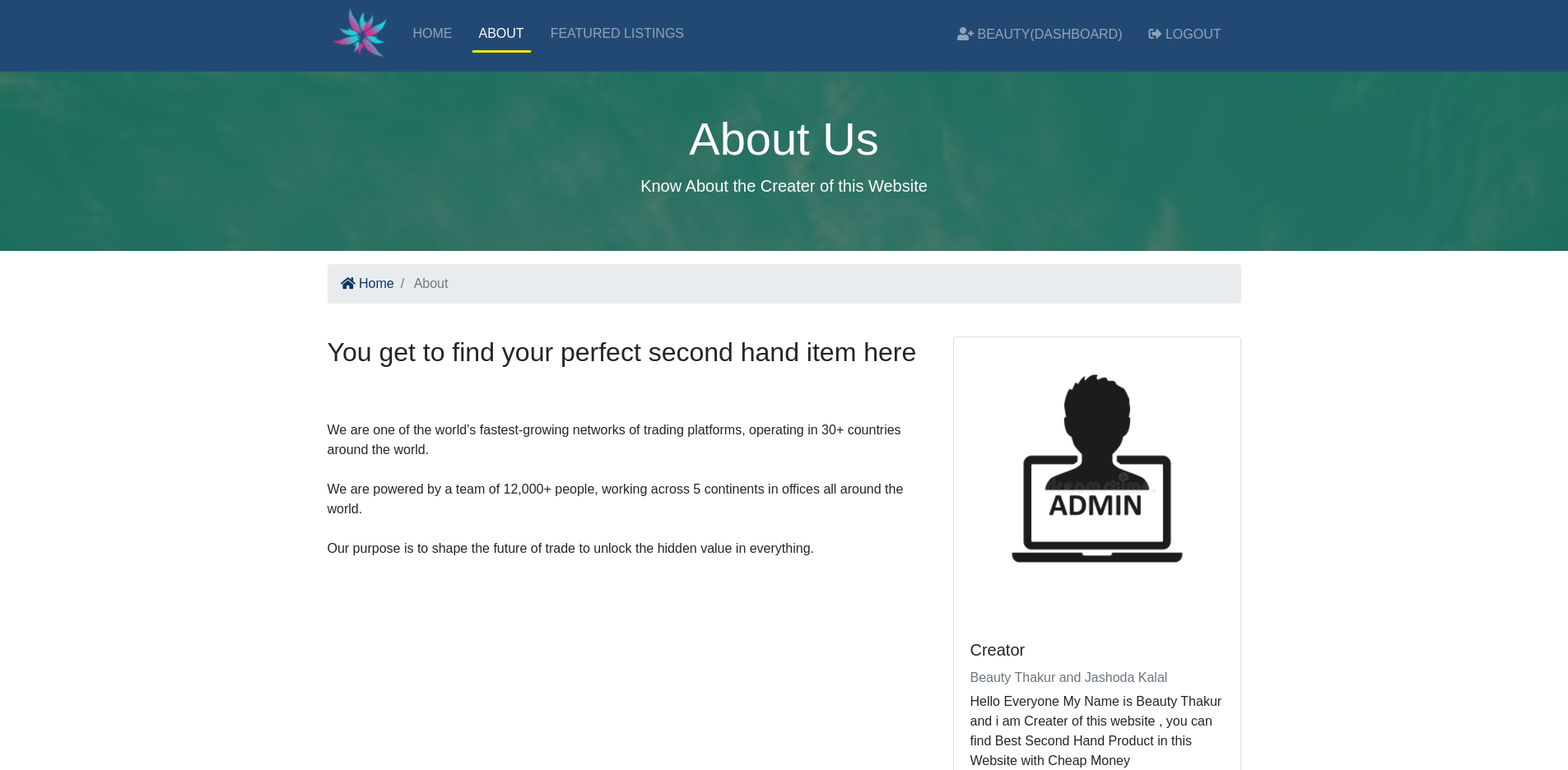
* **Search Page**



**Description : This page is Product Search page. Here to all**

**Category find the product easily.**

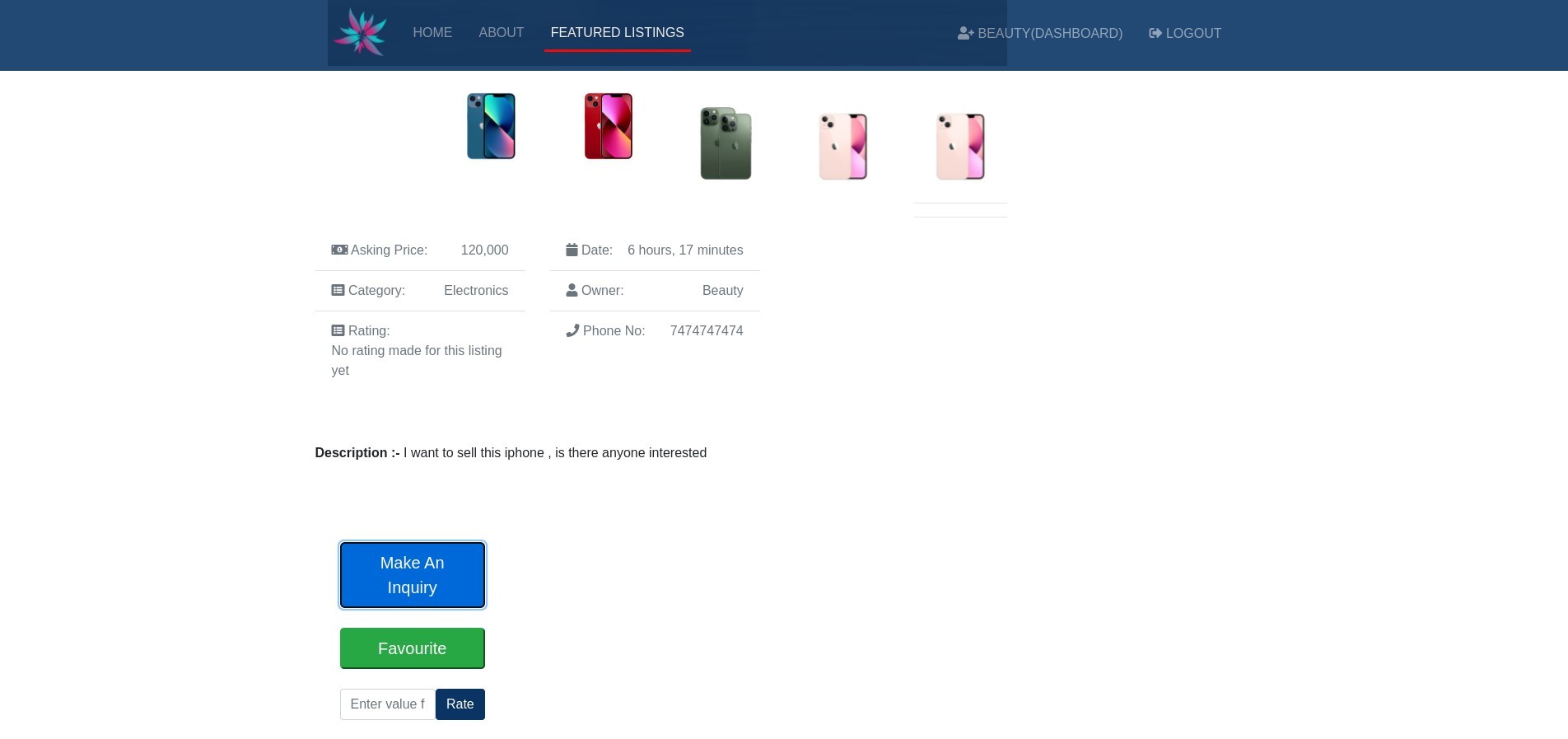
* **About Us**

****

**Description : This page is About page. Here to all**

**Information mention our site.**

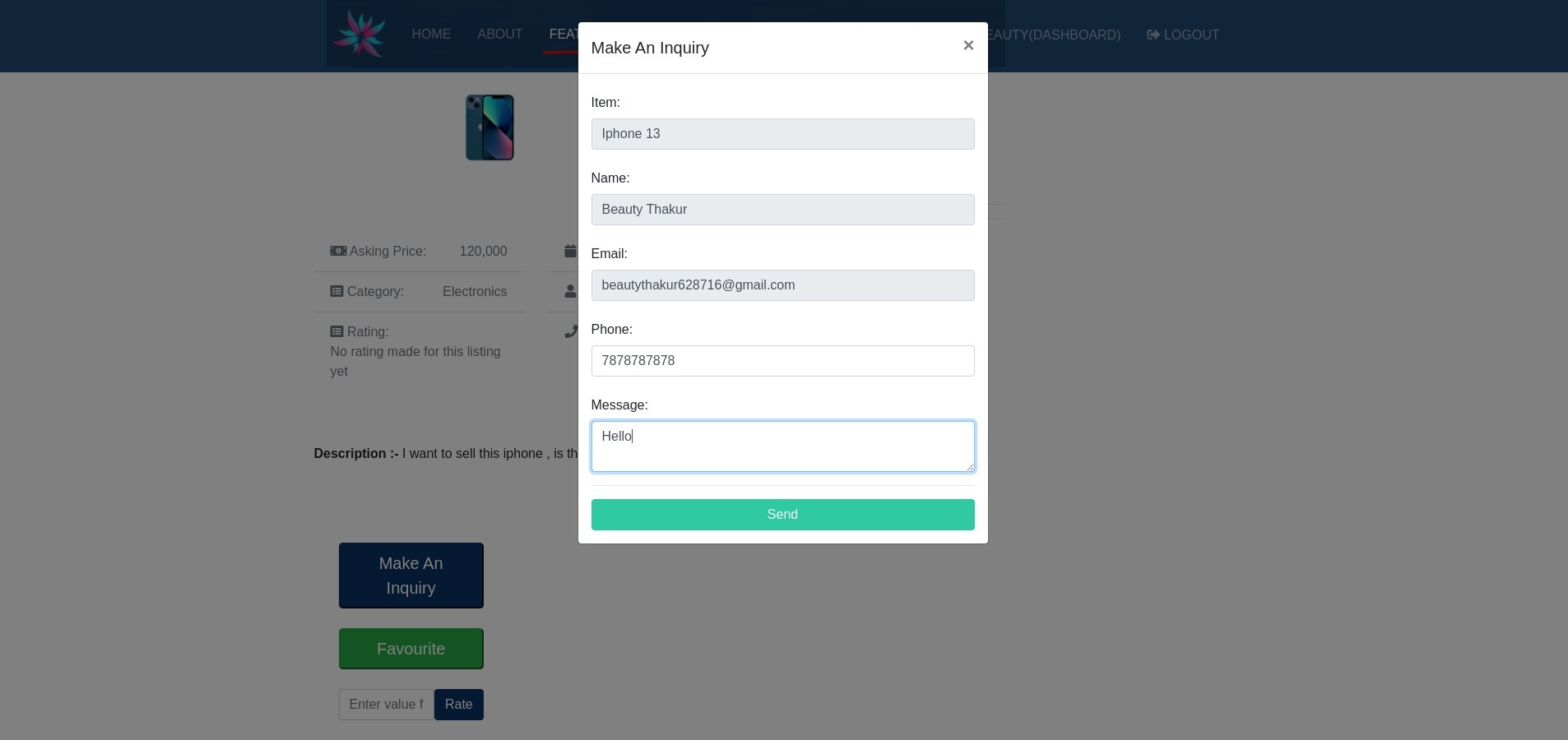
* **Post Detail**



**Description : This page is post Detail here all product detail**

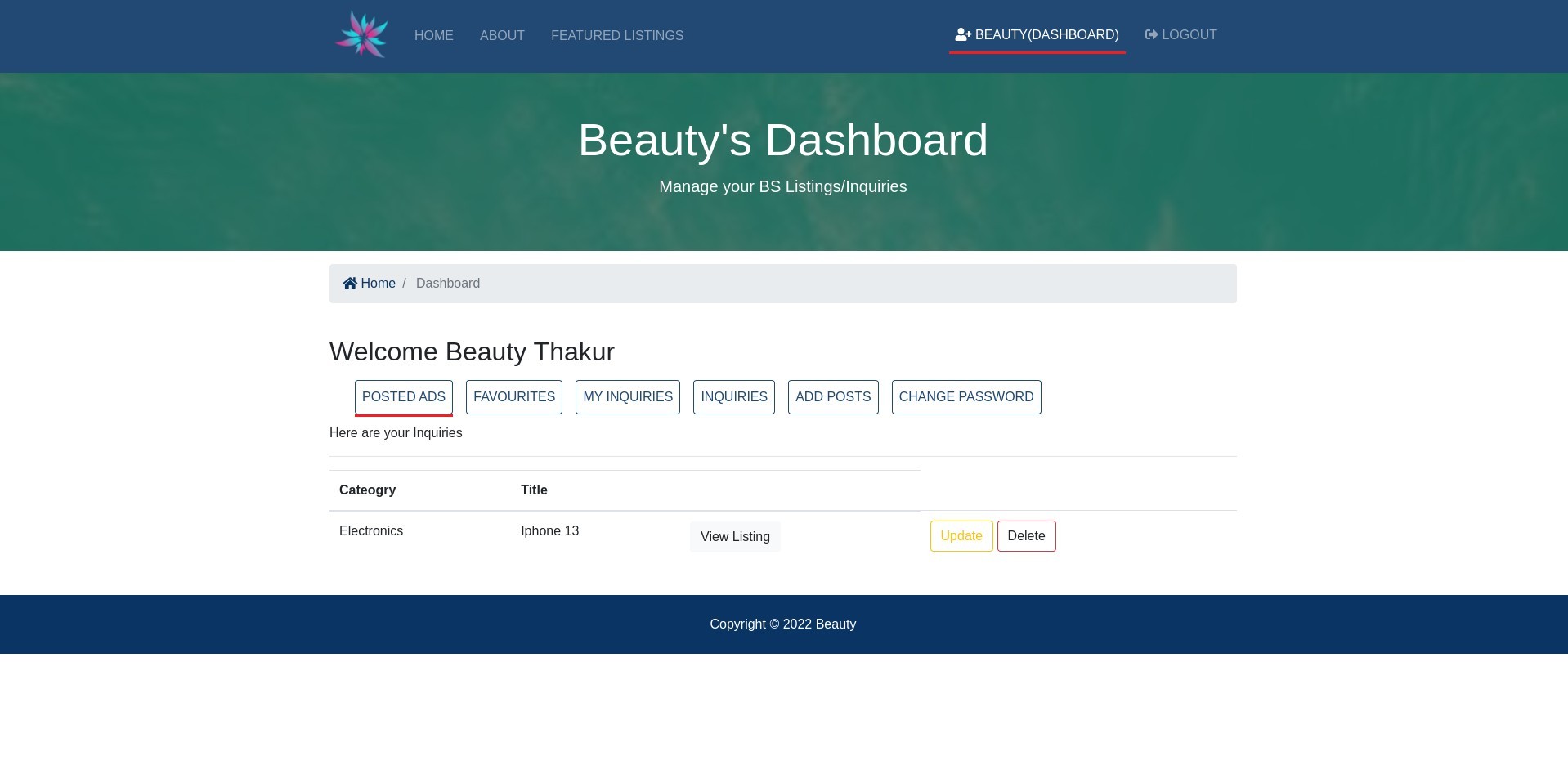
**Mention.**

* **Make an inquiry**



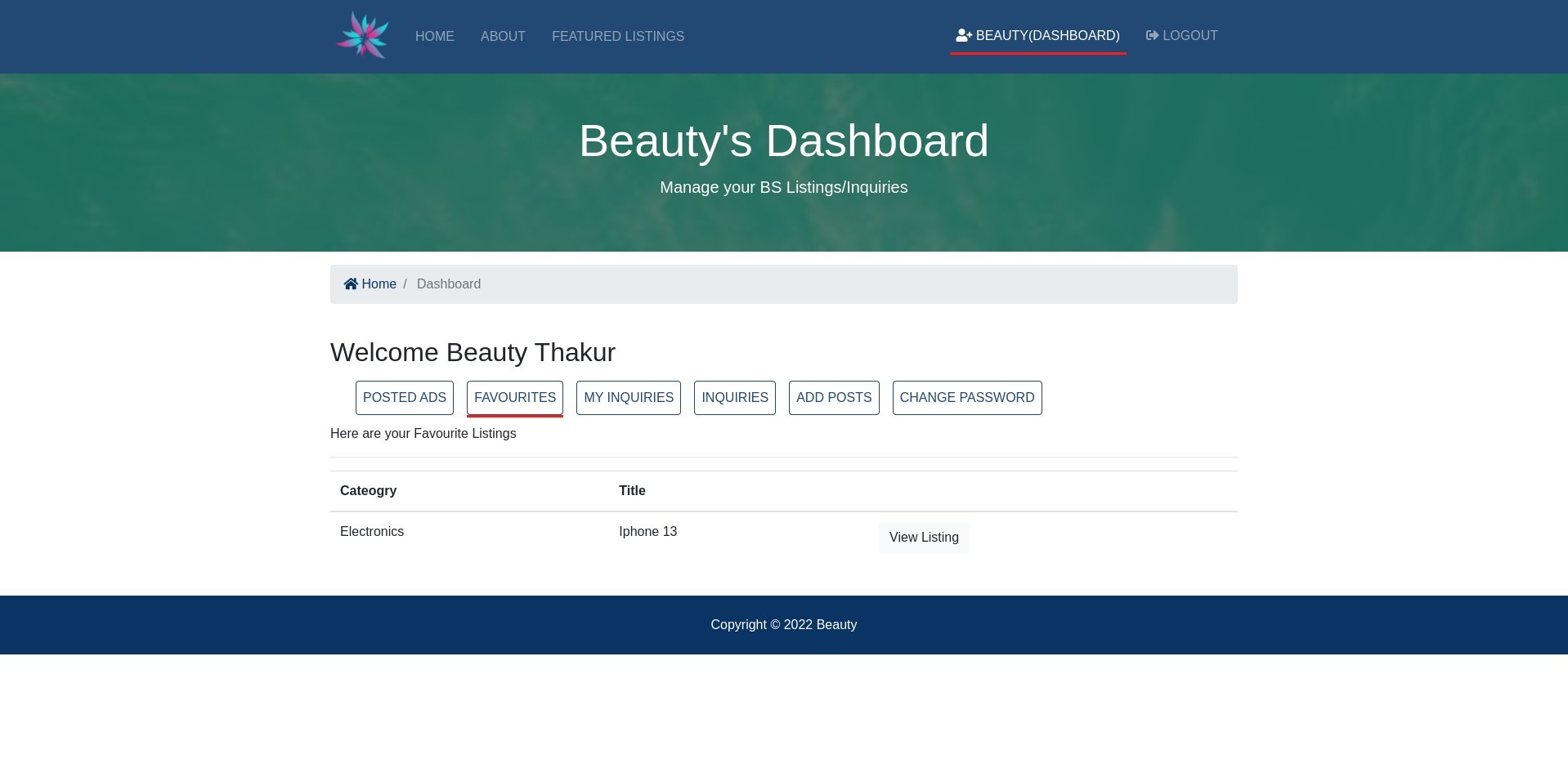
**Description : This page is inquiry product page.**

* **User Dashboard [Posted Ads]**



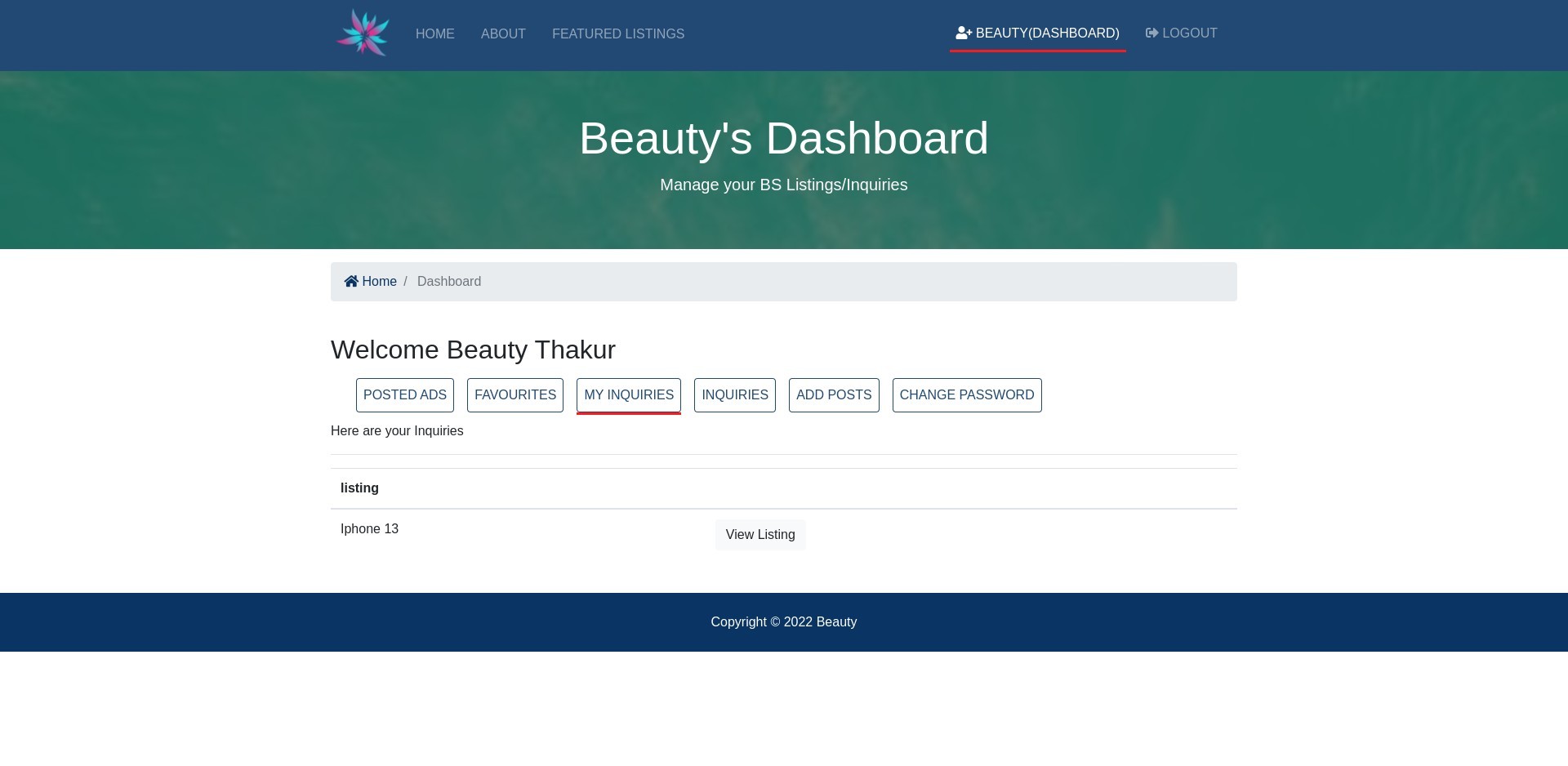
**Description : This page is User dashboard posted ads[Client].**

* **User Dashboard [Favourites]**



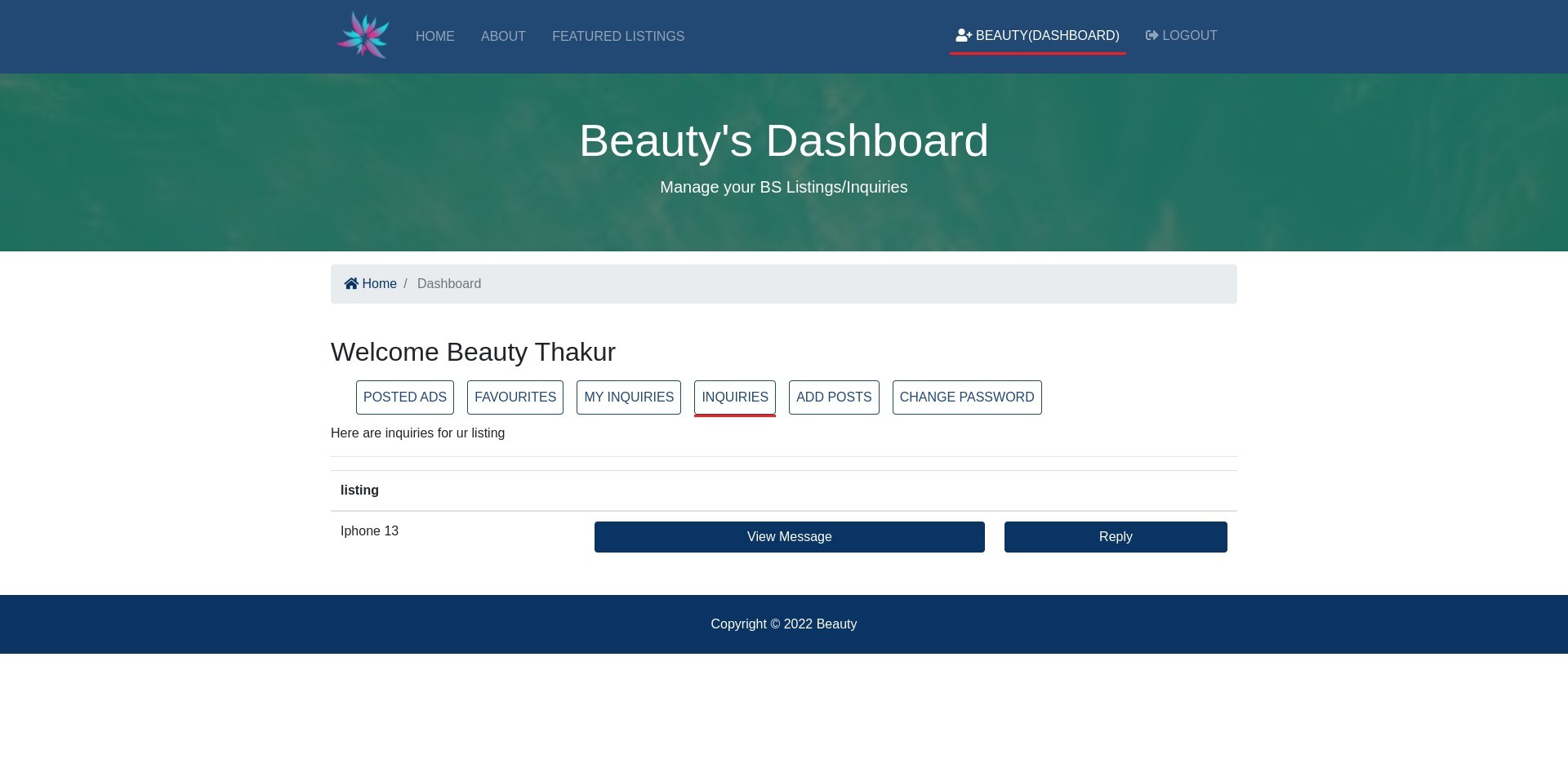
**Description : This page is User dashboard favourite[Client].**

* **User Dashboard [My Inquiries]**



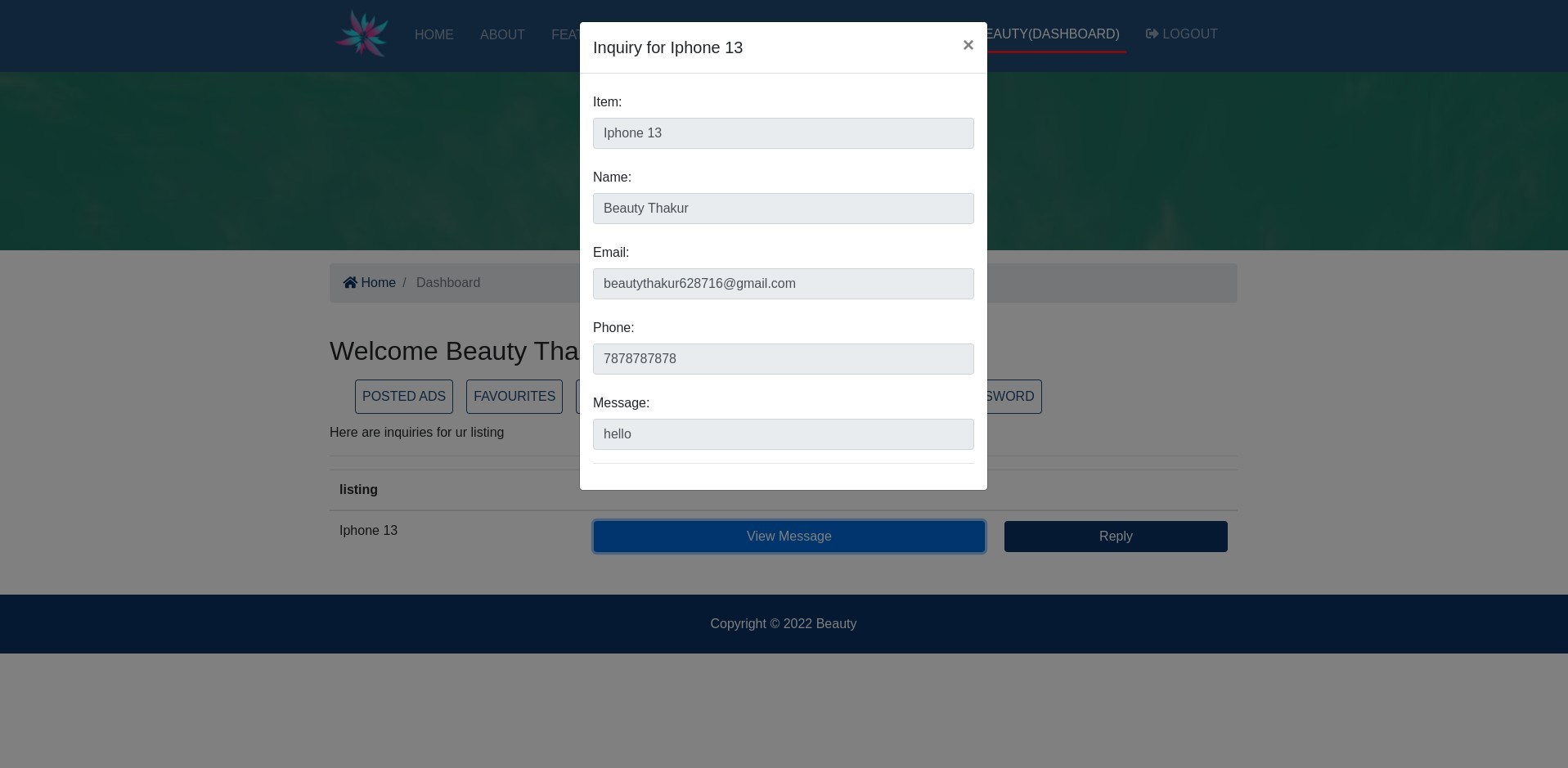
**Description : This page is User dashboard My Inquiries[Client].**

* **User Dashboard [Inquiries]**



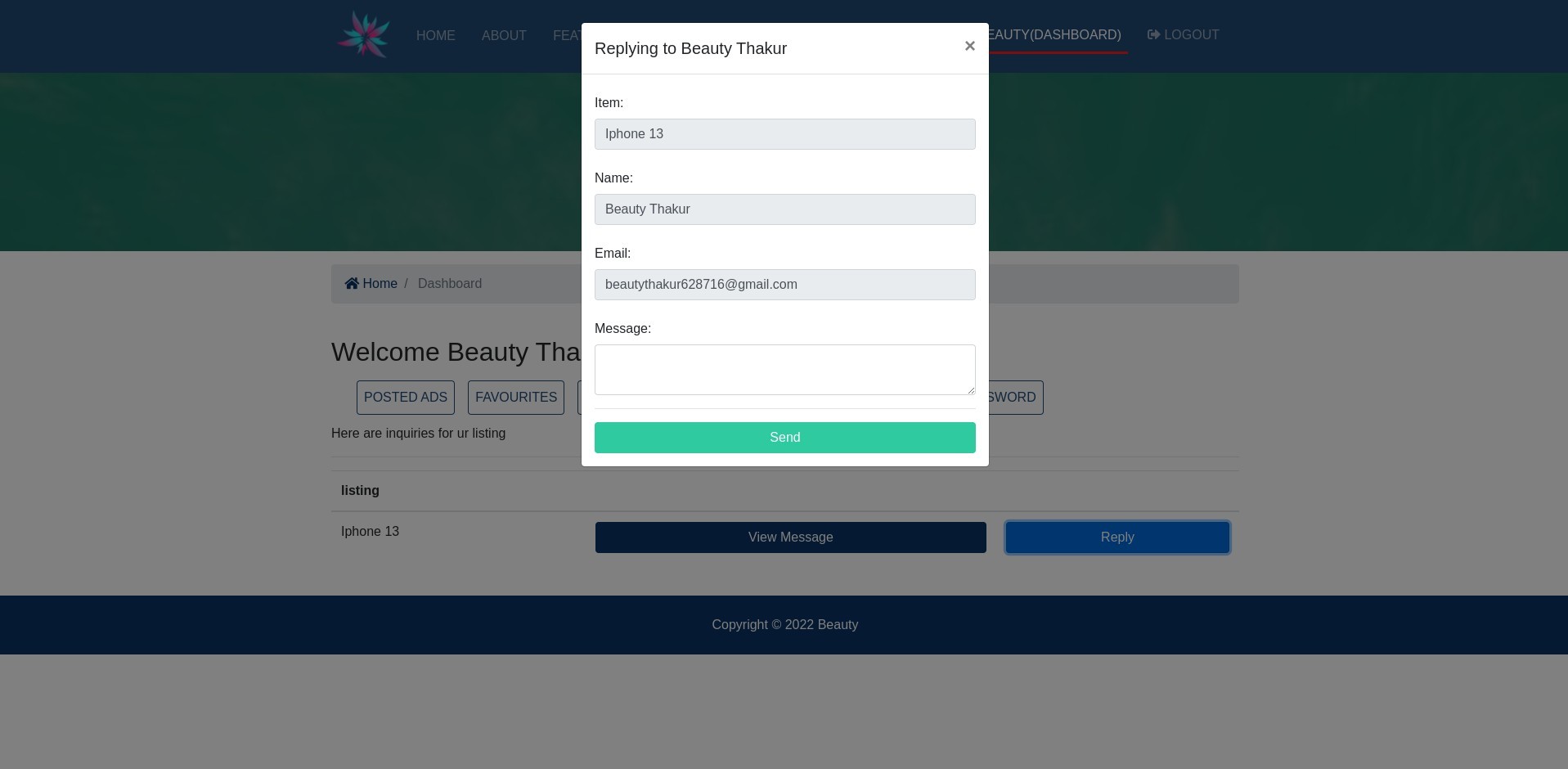
**Description : This page is User dashboard Inquiries[Client].**

* **User Dashboard [Inquiries View]**



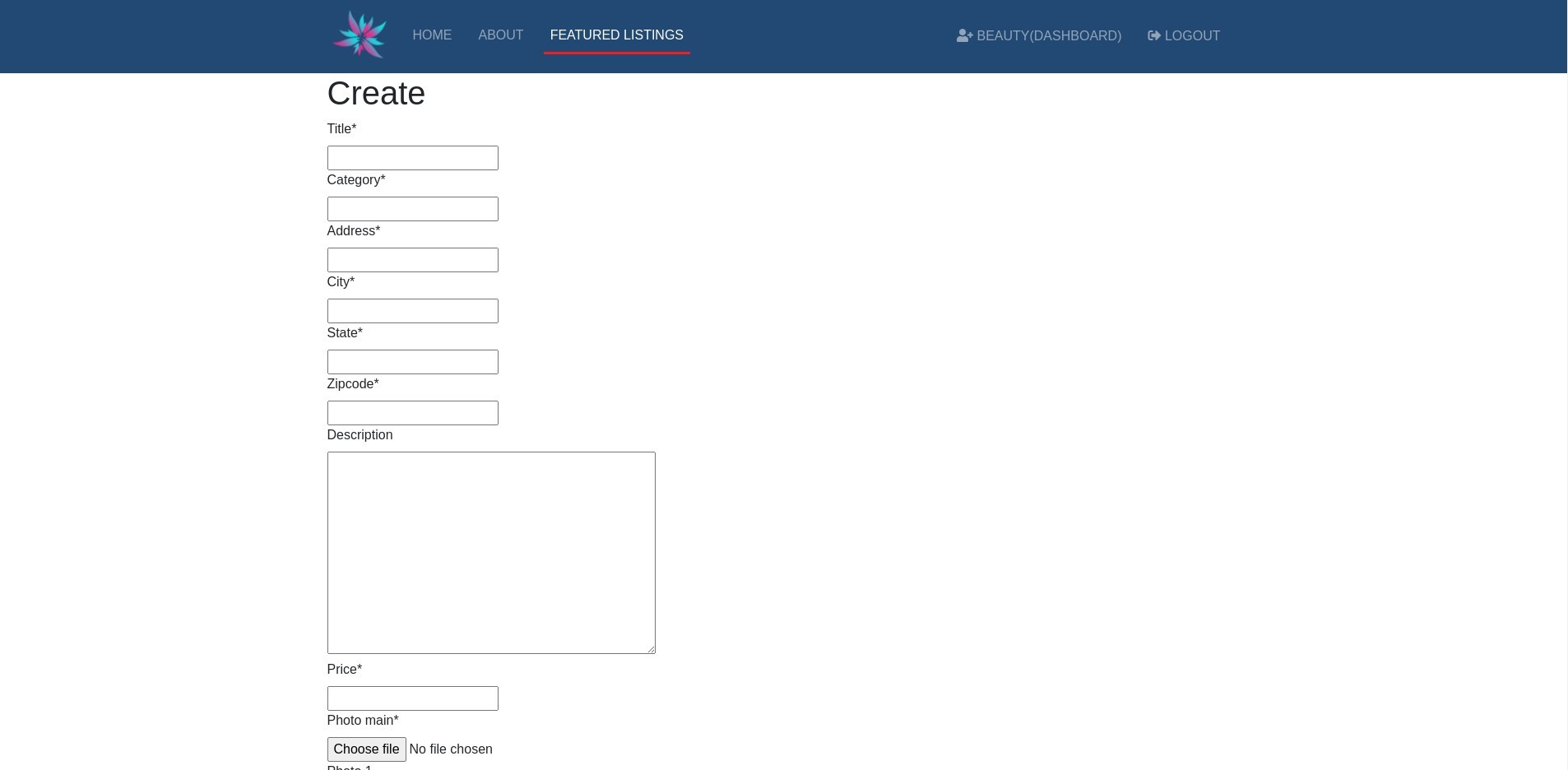
**Description : This page is User dashboard Inquiries View[Client].**

* **User Dashboard [Inquiries Reply]**



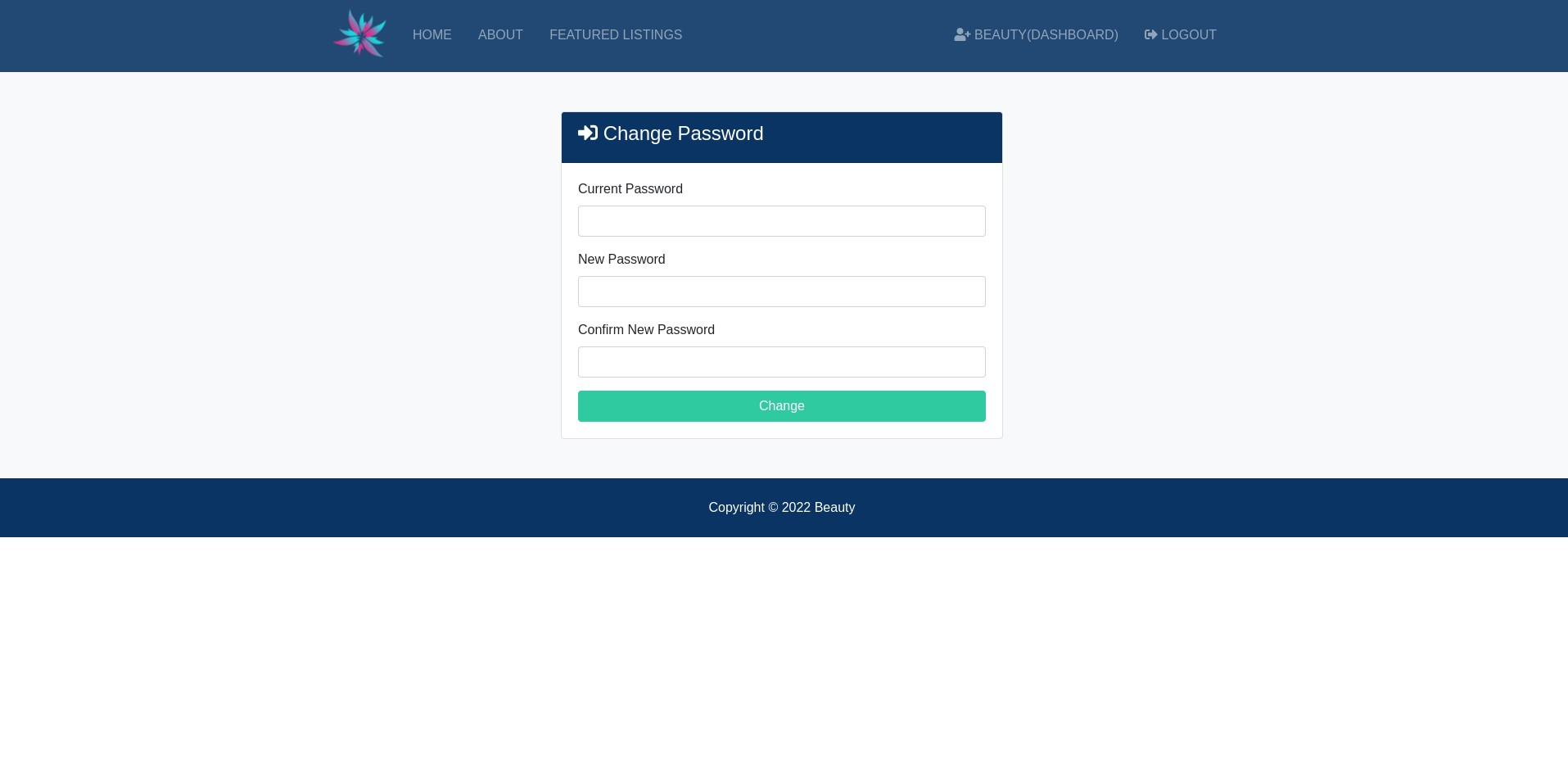
**Description : This page is User dashboard Inquiries Reply[Client].**

* **User Dashboard [Add Post]**



**Description : This page is User dashboard Add Post [Client].**

* **User Dashboard [Change Password]**



**Description : This page is change the current password page.**

**6. TESTING**

#### 6.1 GENERATING TESTCASES

 A test case is a unit of testing activity. Test cases have three parts:

* Goal- the aspect of thesystem being tested.

* Input and System State- data provided to the system under stated conditions.

* Expected Behavior-the output or action the system should take according to these requirements.

##### **1. TESTING STRATEGIES**

**Black-Box Testing:**

Black-box testing, also called behavioural testing, focuses on the functional requirements of the software. That is, black-box testing enables the software engineer to derive sets of input condition that will fully exercise all functional requirements for a program. Black-box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

Black-box testing attempts to find errors in the following categories:

* Incorrect or missing functions.
* Interface errors.
* Errors in data structures or external database access.
* Behaviour or performance errors.
* Initializing and termination errors.

**White-Box Testing:**

White box testing sometimes called glass-box testing is a test design method that uses the control structure of the procedural design to derive

test cases. Using white-box testing methods, the software engineer can derive test cases that...

* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structures to ensure their validity.

#### 6.2 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The unit test focuses on the internal processing logic and data structures within the boundaries of a component. This type of testing can be conducted in parallel for multiple components.

#### 6.3 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the software architecture while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

* **Top-down integration:**

It is an incremental approach to construction of the software architecture. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module.

* **Bottom-up integration:**

It begins construction and testing with atomic modules. Because components are integrated from the bottom up, processing required for

components subordinate to a given level is always available and the need for stubs is eliminated. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system.

* **Recovery Testing:**

It is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed. If recovery is automatic, initialization, check pointing mechanisms, data recovery, and restart are evaluated for correctness. If recovery requires human intervention, the mean-time-to-repair is evaluated to determine whether it is within acceptable limits.

* **Security Testing:**

Security testing verifies that protection mechanisms built into a system will, in fact, protect it from improper penetration. During security testing, the tester plays the role(s) of the individual who desires to penetrate the system. The role of the system designer is to make penetration cost more than the value of the information that will be obtained.

* **Stress Testing:**

Stress testing executes a system in a manner that demands resources in abnormal quantity, frequency, or volume.

* **Performance Testing:**

Performance testing is designed to test the run-time performance of software within the context of an integrated system. It occurs throughout all steps in the testing process. Even at unit level, the performance of an individual module may be assessed as tests are conducted. Performance tests are often coupled with stress testing and usually require both hardware and software instrumentation.

#### 6.4 SYSTEM TESTING

This website can be run on any machine as well as Web Browser like Mozilla, Netscape Navigator and Internet Explorer 6.0 onward.

In this testing I tack the requirements Document of Company functionality, reliability and maintainability.

#### 6.5 ALPHA & BETA TESTING

Alpha and Beta Testing are the part of validation Testing. In this Testing we perform the test according to the both user and developer‘s way. This testing is very useful at the ending stage of the system. And we concentrate on both the way with satisfaction.

Alpha & Beta testing is impossible for the developer without consider the user views and its satisfaction. So in this testing we perform testing twice for user and developer.

* **Alpha Testing:**

Alpha Testing is used in natural application and in common way which every developer performed in system. This testing is conduct in present of developer and user not present at the time of this testing. This technique is generally used to finding the errors.

* **Beta Testing:**

Beta testing is used in live application with advanced way. Here user and developer both are presented and they communicate their views directly to each other. In these techniques we show the system to the user and user test the system with own way.

This technique is generally used to perform some modification in the system which user wants to be changed.

* **Admin Side :**

 **Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Input Data** | **Valid/ Invalid** | **Validation** |
|  |  |  |  |  |
| 1 | Username | Blank | Invalid | Please enter username and password |
|  | Username | Beauty | Invalid | Please enter correct username and password |
|  | Username | Jashoda | Valid | Redirect towards the page. |
| 2 | Password | Blank | Invalid | Please enter username and password |
|  | Password | 123 | Invalid | Please enter correct username and password |
|  | Password | 456 | Valid | Redirect towards the page. |

* **Client Side :**  **Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Input Data** | **Valid/ Invalid** | **Validation** |
|  |  |  |  |  |
| 1 | Email | Blank | Invalid | Please enter username and password |
|  | Email | Beauty12345@gma  il.com | Invalid | Please enter correct username and password |
|  | Email | Jashoda@gmail.ocm | Valid | Redirect towards the page. |
| 2 | Password | Blank | Invalid | Please enter username and password |
|  | Password | 123 | Invalid | Please enter correct username and password |
|  | Password | 456 | Valid | Redirect towards the page. |

## 7. Future Enhancement

**7.1 Future Enhancement**

* We will Authoritative and granular data on the product market in India.
* Data on volume and value changes, brand dynamics and distribution trends in order to effective plan strategies**.**

* + - The software E-commerce has a very user-friendly interface. Thus the users will feel very easy to work on it. The software provides accuracy along with a pleasant interface. Make the present manual system more interactive, speedy and user friendly.
* Make online chat facility for client and developer interaction.

### 8.Bibliography

### Sites referenced :

* + [www.w3schools.com](http://www.w3schools.com/)
  + www.stackoverflow.com
  + https://docs.djangoproject.com/en/4.0/topics/email/