**E-CRAFT**

A Project

Report

Submitted to

**GOVERNMENT POLYTECHNIC COLLEGE, BALAGHAT (M.P.)**

****

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR**

**THE AWARD OF**

**DIPLOMA IN COMPUTER SCIENCE & ENGINEERING**

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA**

**BHOPAL (M.P.)**

**Session: 2023-2024**

**Submitted by**

**1. AAYUSHI DAHERIYA 21012C04002**

**2. AKASH THAKRE 21012C04006**

**3. DHARAMCHAND PATLE 21012C04016**

**4. KHUSHAL NEWARE 21012C04026**

**5. MAMTA DAHARWAL 21031C04029**

**6. MANASVI CHOUDHARI 21012C04030**

**7. PRIYANSH GOSWAMI 21012C04039**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GOVERNMENT POLYTECHINC COLLEGE BALAGHAT**



**Balaghat - 481001[M.P.]**

GOVERNMENT POLYTECHINC COLLEGE BALAGHAT

****

**DECLARATION**

We are the Diploma Student of **Government Polytechnic College, Balaghat, [M.P.]** Department of **Computer Science and Engineering** hereby declares that we own full responsibility for the information. Results and conclusions provided in this project work Titled ***“E-CRAFT”*** submitted to RGPV (Diploma wing) for the award of **Diploma in Computer Science & Engineering** to the best of our knowledge, this project work has not been submitted in part or full elsewhere in any other institution / organization for the award of any certificate/ diploma / degree. We have completely taken care in acknowledging the contribution of others in this academic work. We further declare that in case of any violation of intellectual property rights and particulars declare, found at any stage as the candidate will be solely responsible for the same.

|  |  |  |
| --- | --- | --- |
| **NAME OF STUDENT** | **ROLL NO.** | **SIGNATURE OF STUDENT** |
| **AAYUSHI DAHERIYA** | **21012C04002** |  |
| **AKASH THAKRE** | **21012C04006** |  |
| **DHARAMCHAND PATLE** | **21012C04016** |  |
| **KHUSHAL NEWARE** | **21012C04026** |  |
| **MAMTA DAHARWAL** | **21012C04029** |  |
| **MANASVI CHOUDHARI** | **21012C04030** |  |
| **PRIYANSH GOSWAMI** | **21012C04039** |  |

** GOVERNMENT POLYTECHINC COLLEGE BALAGHAT**

**CERTIFICATE**

This is to certified that the Project work entitled ***“E-CRAFT”*** is a record of work carried out by AAYUSHI DAHERIYA (21012C04002), AKASH THAKRE (21012C04006), DHARAM CHAND PATLE ( 21012C04016 ) , KHUSHAL NEWARE ( 21012C04026 ) , MAMTA DAHARWAL (21031C04029) , MANASVI CHOUDHARI (21012C04030) , PRIYANSH GOSWAMI ( 21012C04039) under our guidance andsupervision for the award of **Diploma in Computer Science & Engineering** at **Government Polytechnic College Balaghat** to the best of our knowledge and belief. The Report: -

1. Embodies the work of candidate himself /herself.
2. Has duly been completed.
3. Fulfil the Partial requirement of RGPV
4. Is up to the standard both in respect of contents and language for being referred to the examiners.

PLACE:- BALAGHAT

DATE: -

**Miss. PRAGYA BALLEY Mrs. VIMLA UIKEY**

Guest Faculty(Dept. of C.S.E.) Guest Faculty(Dept. of C.S.E.)

Co-Guide(Technical) Guide(Technical)

**Miss. UMA GHOSH Mr. R.M. SONWAY**

I/C H.O.D. Principal

(Head of Project Committee) Govt. Polytechnic College

Dept. of Computer Science & Engg. Balaghat

**GOVERNMENT POLYTECHINC COLLEGE BALAGHAT**

****

**CERTIFICATE BY THE EXAMINER**

The Project work entitled ***“E-CRAFT”*** submitted by AAYUSHI DAHERIYA (21012C04002), AKASH THAKRE (21012C04039), DHARAMCHAND PATLE (21012C04016),KHUSHAL NEWARE (21012C04026), MAMTA DAHARWAL (21012C04029), MANASVI CHOUDHARI (21012C04030), PRIYANSH GOSWAMI (2101204039), here by recommended for the award of the **Diploma in Computer Science & Engineering** of **RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA BHOPAL.**

**Sign of Internal Examiner Sign of External Examiner Name: - Name: -**

**Date: - ……………. Date: - ………………**

**Exam venue: -**

**Department of Computer Science & Engineering**

**Government Polytechnic College**

**Balaghat 481001 (M.P.)**

**ACKNOWLEDGEMENT**

We have taken of efforts to complete this project ***“E-CRAFT”*** within a stipulated time. However, it would not have been possible without the kind support and help of many individuals and organization. We would like to extend our sincere thanks to all of them.

We are highly indebted to the entire project committee **I/C H.O.D. Miss. UMA GHOSH** (Head of Project Committee), **Mr. SHARAD DAHATE,** **Mrs. VIMLA UIKEY, Mrs. TRAPTI AADKANE, Miss. LAXMI THAKUR, Miss PRAGYA BALLEY, Mrs. RITU SONI** for their constant guidance, supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We are also obliged to extend my thanks to respected principal sir **Mr. R. M. SONWAY** for providing all the required amenities.

We would like to express our gratitude towards the entire faculty member, Technical and Nontechnical staff of **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** for their kind cooperation and encouragement which helps us in completion of this project.

We would like to express our special gratitude and thanks to our reverential parents for their constant inspiration of emotional support to complete this project. We would like to extend our special thanks to industry person for giving us such attention and time. Last but not least we would like to thanks to all the friends and people who have not only willingly helped us out their abilities but also constantly extended their support by suggestion as well as criticism during development of the project.

PLACE: - BALAGHAT

DATE: -

|  |  |  |
| --- | --- | --- |
| **NAME OF STUDENT** | **ROLL NO.** | **SIGNATURE OF STUDENT** |
| **AAYUSHI DAHERIYA** | **21012C04002** |  |
| **AKASH THAKRE** | **21012C04006** |  |
| **DHARAMCHAND PATLE** | **21012C04016** |  |
| **KHUSHAL NEWARE** | **21012C04026** |  |
| **MAMTA DAHARWAL** | **21012C04029** |  |
| **MANASVI CHOUDHARI** | **21012C04030** |  |
| **PRIYANSH GOSWAMI** | **21012C04039** |  |

**ABSTRACT**

The main objective of ***‘’E-CRAFT’’*** Commercial Website is to enhance the art of local area artisan and they have to provide work through this platform. This platform is a simple and easy for their work and to fill their arts.

The objective of the craft website is to make various artisan of the local or area and their arts accessible to the people through the website.

This website will have many crafts like bamboo craft, metal craft, jute craft etc. which the user can purchase and sell. Through this website, the artisan can work within the NGO and the user can also create his own shop and sell the product there.

Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to use his skill so that he can benefit from it. In the platform, work will be provided to the users among NGOs.

Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to practice his skills so that he can benefit from it. In this craft, work will be provided to the users through the NGO.

**CONTENT**

1. **INTRODUCTION…………………………………………………………………………………………....(2-3)** 
   1. Introduction…………………………………………………………………………........(2)
   2. Objective……………………………………………………….…………………….…..(2-3)
   3. Advantages……………………………………………………………..………..…………(3)
2. **LITERATURE SURVEY………………………………………………………………………………… (5-11)**
   1. Technologies……………………………………………………….…………………(5-6)
   2. Software Development Life Cycle…………………………………...…(7)
   3. Agile Model…………………………………………………………………………..(8-9)
   4. Market Survey…………………………………………………………...……...(10-11)
3. **ANALYSIS…………………………………………………………………………………………...……………… (13-14)**
   1. Function Requirements…………………………………………………..……(13)
   2. Non-Functional Requirements………………………………………..…(13)
   3. Hardware Requirements……………………………………………………...(13)
   4. Software Requirements…………………………………………….…………(13)
   5. Feasibility Study……………………………………………………………...(13-14)
4. **DESIGN** **……………………………………………………………………………………….………………............ (16-21)**
   1. E-R Diagram………………………………………………………….………..…16-17)
   2. Use Case Diagram……………………………………………………………..…(18)
   3. Data Flow Diagram………………………………………...………………(19-21)
5. **CODING………………………………………………………………………………………………….…..…......... (21-45)**
6. **TESTING……………………………………………………………………………………………………...……….(47-51)**
   1. White Box Testing……………………………………………………….……….(48)
   2. Black Box Testing……………………………………………………………..…(48)
   3. Gray Box Testing ………………………………………………..................……(49)
   4. Testing Levels……………………………………………...…………...…...(49-51)
7. **OUTPUT SCREEN………………………...……………………………………………………….….. (51-57)**
8. **COST ESTIMATION…………………………………………………………………………………(59-60)**
9. **CONCLUSION………………………………………………………………………………………..………... (62)**
10. **BIBLIOGRAPHY……………………………………………………………………………………….…… (64)**

#### **LIST OF DIAGRAMS**

1. **SDLC……………………………………………………………………………………………………….……………… (7)**
2. **Agile Model………………………………………………………………………………………..……………………(8)**
3. **ER Diagram ……………………………………………………………………………………..…..………………(17)**
4. **Use Case Diagram ………………………………………………………………………….…………………(18)**
5. **Data Flow Diagram ….…………………………………………………………….…………………..(19-21)**
   1. Level-0………………………………………………………………………………………………….………...…… (19)
   2. Level-1………………………………………………………………………………………………….……..….…… (20)
   3. Level-2………………………………………………………………………………………………….…….……..… (21)

**6. Testing ……………………………………………………………………………………………………..………...(47-51)**

* 1. White Box Testing ………………………………………………………………………………... ….(48)
  2. Black Box Testing… ………………………………………………………………………………… ..(48)
  3. Gray Box Testing……………………………………………………………………………………… ...(49)

1. **Output Screen……………………………………………………………………...………………………….(53-57)**

**CHAPTER-1**

#### 

#### **1.INTRODUCTION**

The main objective of E-Craft Commercial Website is to enhance the art of local area artisan and they have to provide work through this platform. This platform is a simple and easy for their work and to fill their arts.

The objective of the craft website is to make various artisan of the local or area and their arts accessible to the people through the website. This website will have many crafts like bamboo craft, metal craft, jute craft etc. which the user can purchase and sell.

Through this website, the artisan can work within the NGO and the user can also create his own shop and sell the product there. Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to use his skill so that he can benefit from it. In the platform, work will be provided to the users among NGOs.

Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to practice his skills so that he can benefit from it. In this craft, work will be provided to the users through the NGO.

##### **OBJECTIVE: -**

* E-craft website is a commercial platform, it is an online marketplace through which communication will be done between artisans and NGOs.
* Through the website, artisans can sell their crafts through this platform.
* Through e craft website, the user will be able to interact with the customer and through the craft or product of artisan and that is, the high-quality craft product will be shown to the user and the complete information details of the product will be shown.
* First of all, the artisan will have to login to the E-craft website so that he can sell his craft and the login steps are as follows: -

1.Username

2.Password

* After logging in to the E-craft website, you will create a shop to show your craft on the website

**\*** The step of creating shop: -

1. Username

2. Address

3. E-mail ID

4. Craft work

5. Display craft in website

6. Information’s/Details, Price

* Admin will analyser the shop and provide security to them through shop analyser.
* Admin will calculate the profit and tell the cost of shop craft to the shopper so that he can see his profit.
* The selling price and cost of the product will be filled by the shopper through the profit calculators bad listing fee transaction fee processing fee Like website shopper will get 12% product free and customer or shopper will get 88% profit.
* Through this the shopper will be able to add his/her product per website. This means the shopper can add his/her 10 products or craft for free on the website. If there are more than 10 products, there will be a charge for using the product. He/she will have to pack monthly.
* In the e-craft website, the website's own warehouse will be provided to store the artisan's craft product so that the product can be sent to its right location. There will be transaction fee in product stories which will be 0.1% per product.
* Before ordering in the website, the user must sign up. When a user places an order, a notification is sent to the admin of E-craft in which all the information of the user like name, email address, mobile number etc. goes to the admin. After this process the user can confirm his order.
* When the user's order is confirmed, it is sent to the user's address and the product is sent by the delivery boy.

**Advantages: -**

* The objective of the craft website is to make various artisan of the local or area and their arts accessible to the people through the website. This website will have many crafts like bamboo craft, metal craft, jute craft etc. which the user can purchase and sell.
* Through this website, the artisan can work within the NGO and the user can also create his own shop and sell the product there. Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to use his skill so that he can benefit from it. In the platform, work will be provided to the users among NGOs.
* Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to practice his skills so that he can benefit from it. In this craft, work will be provided to the users through the NGO.

**CHAPTER-2**

**2. LITERATURE SURVEY**

## **2.1 TECHNOLOGIES: -**

### **a. HTML: -**

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. .HTML is the language used to create Webpages. "Hypertext" refers to the hyperlinks that an HTML page may contain. "Mark-up language" refers to the way tags are used to define the page layout and elements within the page. The first line defines what type of contents the document contains. "" Means the page is written in HTML5. Properly formatted HTML pages should include, and tags, which are all included in the example above. The page title, metadata, and links to referenced files are placed between the tags. The actual contents of the page go between the tags.

**b. CSS: -**

CSS stand for Cascading Style Sheets.CSS handles the look and feel part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the mark-up languages HTML or XHTML

## **c. JAVA SCRIPT: -**

JavaScript is a scripting language developed by Netscape Navigator to enable web author interactivity sites. Although it shares many of the features and structures of the full java language, it was developed independently. JavaScript can interact with HTML source code, enabling web authors to spice up their sites with dynamic content. JavaScript is endorsed by a number of software companies and is an open language that anyone can use without purchasing a license. It is supported by recent browsers from Netscape and Microsoft, though internet explorer supports only a subset, which Microsoft calls JavaScript. Scripts return with JavaScript can be embedded into HTML documents.

**d. DJANGO: -**

Django is a web framework written in Python that simplifies the development of web applications by providing a set of tools and conventions. Django helps developers build web applications quickly and efficiently by providing pre-built components and following the "Don't Repeat Yourself" (DRY) principle.

**\* Django consists of several key components:**

**• ORM (Object-Relational Mapping):** Allows developers to interact with databases using Python objects.

**• URL Routing:** Maps URLs to views, which are Python functions or classes responsible for processing requests and returning responses.

**• Templates:** HTML files combined with Python variables to generate dynamic web pages.

**• Forms:** Simplifies the handling of HTML forms and form data.

**• Admin Interface:** Automatically generates an admin panel for managing site content.

Overall, Django simplifies web development by providing a structured framework with built-in components, allowing developers to focus on building their applications rather than reinventing the wheel.

**e. PYTHON: -**

Python is a high-level, interpreted programming language known for its readability and versatility, widely used in web development, data analysis, artificial intelligence, scientific computing, and automation. It emphasizes code readability with its notable use of significant indentation.

Python is a computer programming language often used to build websites and software, automate tasks, and analysed data. Python is a general-purpose language, not specialized for any specific problems, and used to create various programmes.

**f. SQLite: -**

SQLite is an embedded, server-less relational database management system. It is an in-memory open-source library with zero configuration and does not require any installation. Also, it is very convenient as it's less than 500kb in size, which is significantly lesser than other database management systems.

An SQLite database requires little or no administration, SQLite is a good choice for devices or services that must work unattended and without human support. SQLite is a good fit for use in cell phones, PDAs, set-top boxes, and/or appliances.

SQLite is a lean software library or database that is integrated into numerous applications. One of its distinguishing features is its lack of a client-server structure. SQLite alternatives include MySQL, MongoDB and PostgreSQL.

### **2.2 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC MODEL):** -

Software Development Life Cycle (SDLC) is a systematic process for building software that ensures the quality and correctness of the software built. The system development should be complete in the pre-defined time frame and cost. SDLC consists of a detailed planned which explains how to plan, built, and maintain specific software. Every phase of the SDLC life cycles its own process and deliverable that fit into the next phase.

* Phase 1: Requirement Gathering and Analysis
* Phase 2: Design
* Phase 3: Coding
* Phase 4: Testing
* Phase 5: Deployment
* Phase 6: Maintenance

Requirement Gathering

and

Analysis

Desi

gn

Coding

Maintenance

Deployment

T

esting

**SDLC**

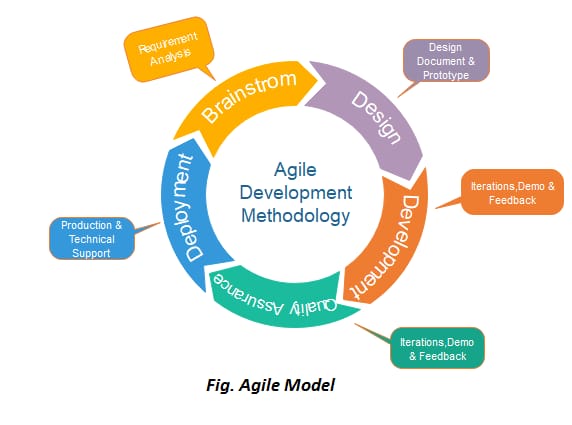
**M**

**ODEL**

**Diagram: - Software Development Life Cycle Model**

**2.3 Agile model: -**

The meaning of Agile is swift or versatile. “Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client

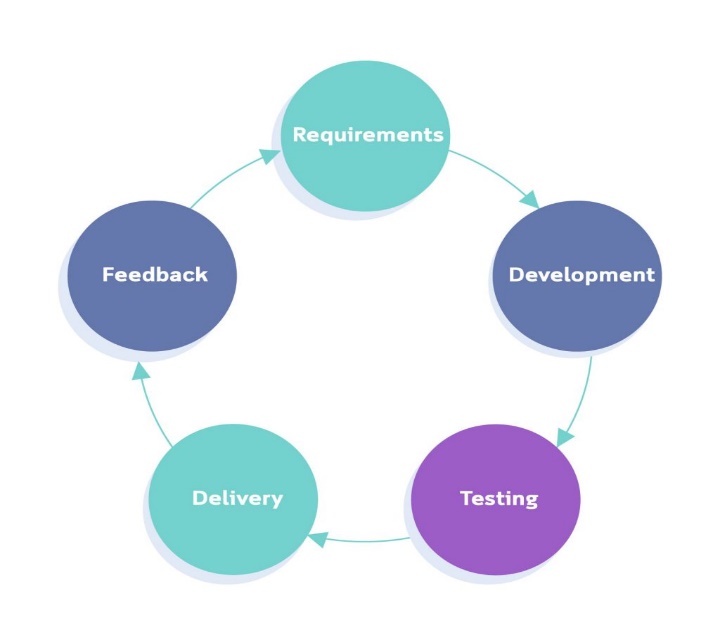
**When we use the Agile Model? -**

* Using an agile model allows companies to implement changes much faster.
* An agile model also allows a company to adapt to changes more quickly.
* Using a traditional methodology, a project's scope is set at the planning stage, and it is difficult to make major changes to it throughout the development process.

**Advantages of the Agile Model: -**

* Customer satisfaction is rapid, continuous development and delivery of useful software.
* Customer, Developer, and Product Owner interact regularly to emphasize rather than processes and tools.
* Product is developed fast and frequently delivered (weeks rather than months.)
* A face-to-face conversation is the best form of communication.
* It continuously gave attention to technical excellence and good design.
* Daily and close cooperation between business people and developers.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed.

**Disadvantages of the Agile Model: -**

* It is not useful for small development projects.
* There is a lack of intensity on necessary designing and documentation.
* It requires an expert project member to take crucial decisions in the meeting.
* Cost of Agile development methodology is slightly more as compared to other development methodology.
* The project can quickly go out off track if the project manager is not clear about requirements and what outcome he/she wants.

 **3. MARKET SURVEY**



**CHAPTER-3**

**3. ANALYSIS**

## **1. FUNCTIONAL REQUIREMENTS: -**

This facility is given to the user to login to the system. They will have to enter the user id and password before they are allowed to enter the system. user id and password will invalid id then user is not allowed to login into the system.

User ID is provided while registering the system should allow only valid ID and password to enter the system. the system performs the anionization process of access a user can have.

## **3. NON-FUNCTIONAL REQUIREMENTS: -**

When library management system is implemented, the librarian and book transactions will be much faster. The system must register correctly. (Example report generation) book transaction and search.

The system has been designed to be user friendly, so that different tasks can be done easily and efficiently in the form of manuals. In implementing the whole system, it is front end with PHP using HTML as its server-side scripting language. Which will be done for database connectivity. the database part is developed using MYSQL.

## **3. HARDWARE REQUIREMENT: -**

* Ram: 512 MB or Higher.
* Processor: Dual core or Higher
* Hard disk: 20 GB or Higher

## **4. SOFTWARE REQUIREMENT: -**

* Operating system: WINDOWS 7/8/10/11/12
* Client-side scripting language: HTML, CSS, JAVA SCRIPT, PYTHON
* Database: SQLite Data Base

**2. FEASIBLITY STUDY: -**

**2.1 TECHNICAL FEASIBLITY: -**

We concern here with specifying equipment and software that will satisfy the user requirement. It will run any platform (machine). It will run with minimum system requirements and with minimum system resources acquired during run. New module can be added later on the application, if required in the future.

**2.2 OPERATIONAL FEASIBLITY: -**

The system will be easy to use as user interface is GUI based. The system is easy to use so no any special skills will be required to use the system. New user will find it easy to use. So, the project will be operationally feasible.

**2.3 ECONOMIC FEASIBLITY:-**

The procedure is to determine the benefit and saving that are expected from the project and compare them with the cost. As internet as the cheapest way of communication, we can perform communication using web. The cost is just the cost of using the internet based on the channel allocation.so the project will be economical feasible.

**2.4 SOCIAL FEASIBLITY: -**

The project will be socially feasible as today user want quick services in everywhere in a large geographical area in feasibility study face, we had undergone though various steps which are describe: Identify the exception of user from computerized system.

**CHAPTER-4**

**4.DESIGN**

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

**Here we have used the following diagrams to design the software**:

4.1 ER Diagram

4.2 Use Case Diagram

4.3 Data Flow Diagram

### **4.1 E-R DIAGRAM: -**

The name of the E-R model is the Entity Relationship Model. It is a high-level data model. This model has been used to define the data elements and relationship for a system in other words, "In DBMS, ER model is a data model that describes the structure of the database with the help of diagrams”.

**Component of E-R Diagram**

The E-R model is also called the E-R diagram because it presents the entities in the form of a diagram (picture) and shows the relationship between the entities. This model was developed by Peter in 1976. The E-R model is used to present the conceptual schema of the real-world.

There are three main components of the ER model which are as follows: -

a. Entity

1. Relationship
2. Attribute

**Entity: -**

An entity can be any person, place, car and real word object. In the ER diagram, an entity is

represented by a rectangle. Entity must have an attribute and a unique key.

**Entity set: -**

An entity set is a group of entities of the same type that share similar properties. Example of this – Student, Teacher, Class and Course can be considered as entities in a school database. If a student is an entity, then the datasets of all the students are called entity set.

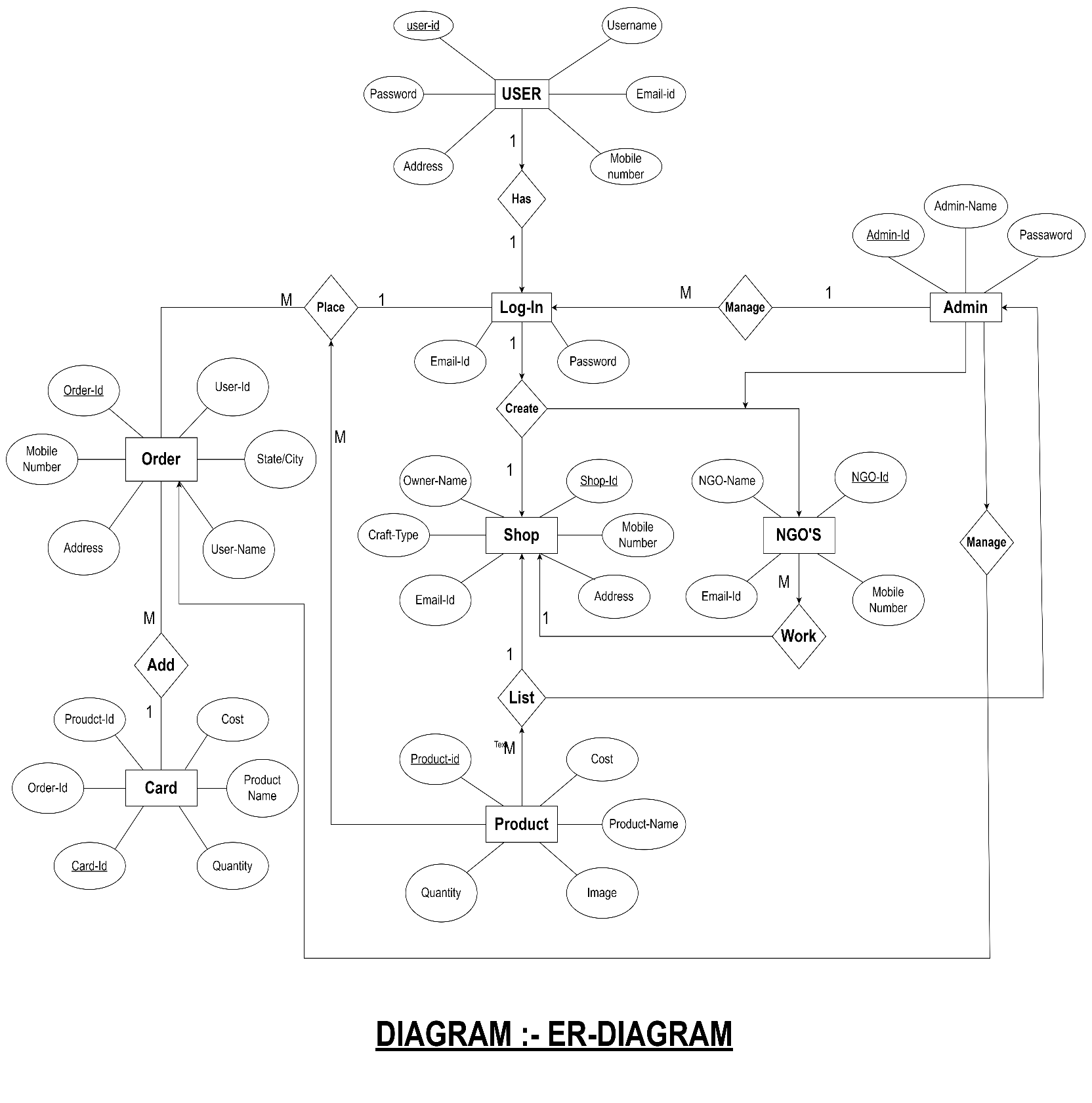
**Relationship: -**

Relationship is used to describe the relation between entities. It is represented by a diamond.

**For example: -**  Teacher teaches at school and soldier enrols in a military. Here leachate and enrols are called relationship.

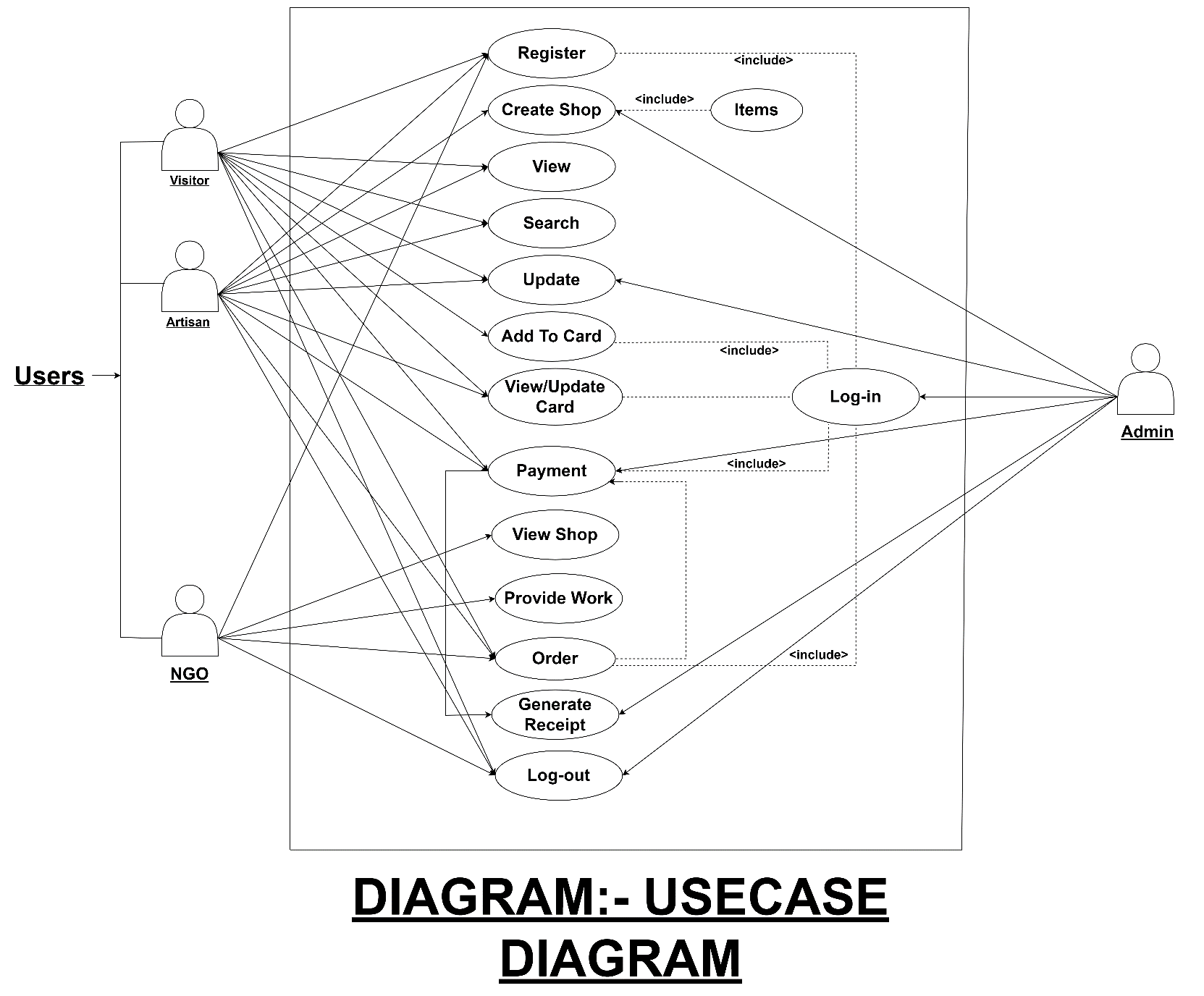
#### **Attribute: -**

Attribute is used to describe the property of an entity. It is represented by oval. For example- student is an entity and its subject name, subject code and gender are it



**4.2 USE CASE DIAGRAM: -**

Use Case Diagram capture the system’s functionality and requirements by using actors and use case. Use case represent high-level functionalities and how a user will handle the system. Use-cases are the core concepts of Unified Modelling language modelling. A use case diagram should be simple and contains only a few shapes. If yours contain more than 20 use cases, you are probably missing use case diagram. Use cases represent only the functional requirements, and implementation constraints must be represented separately, again, with other UML diagrams.

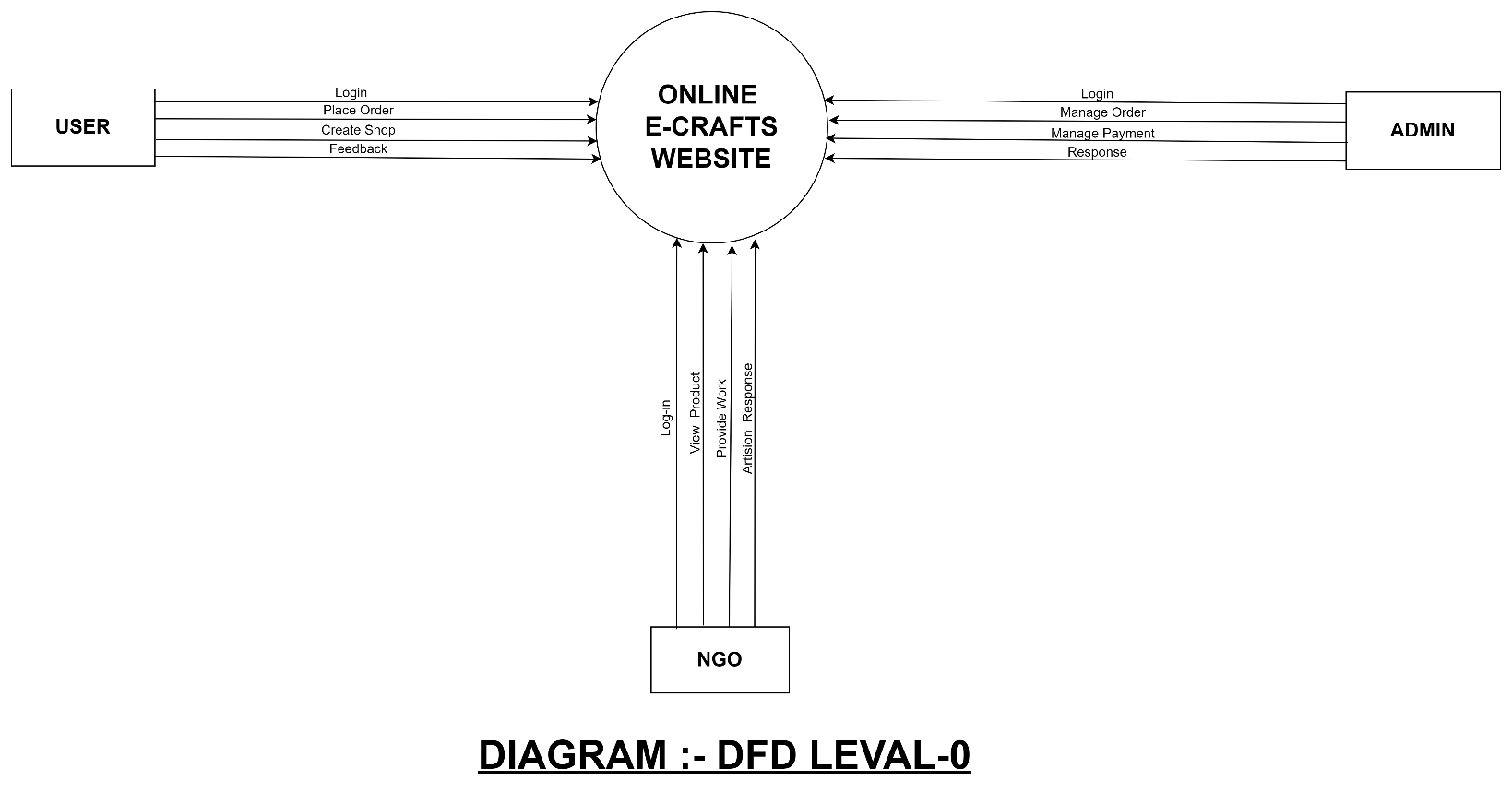


#### **4.3 DATA FLOW DIAGRAM: -**

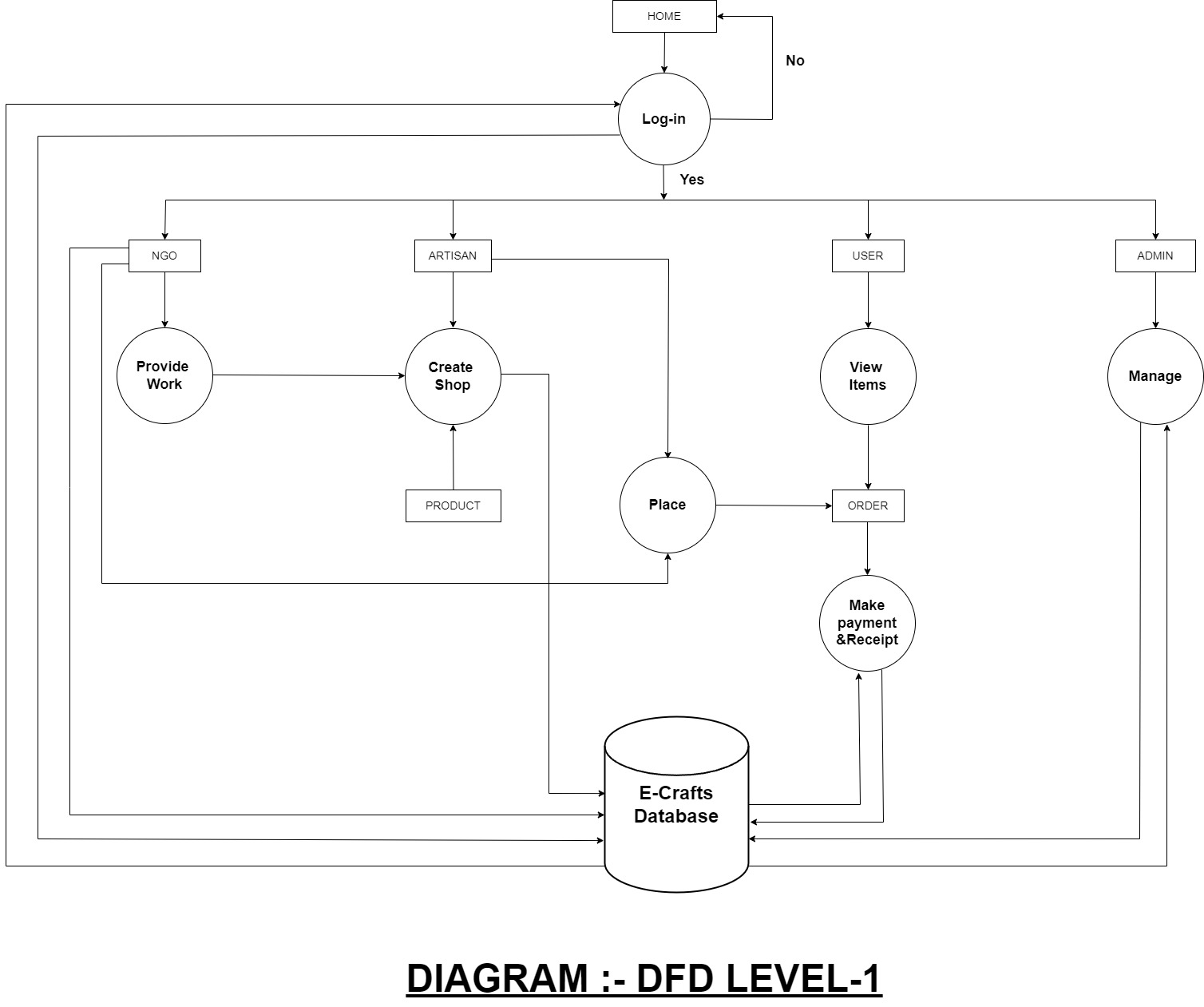
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD.It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analysing existing as well as proposed system.Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modelling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes

**Levels in DFD: -**

##### **a. DFD Level 0: -**

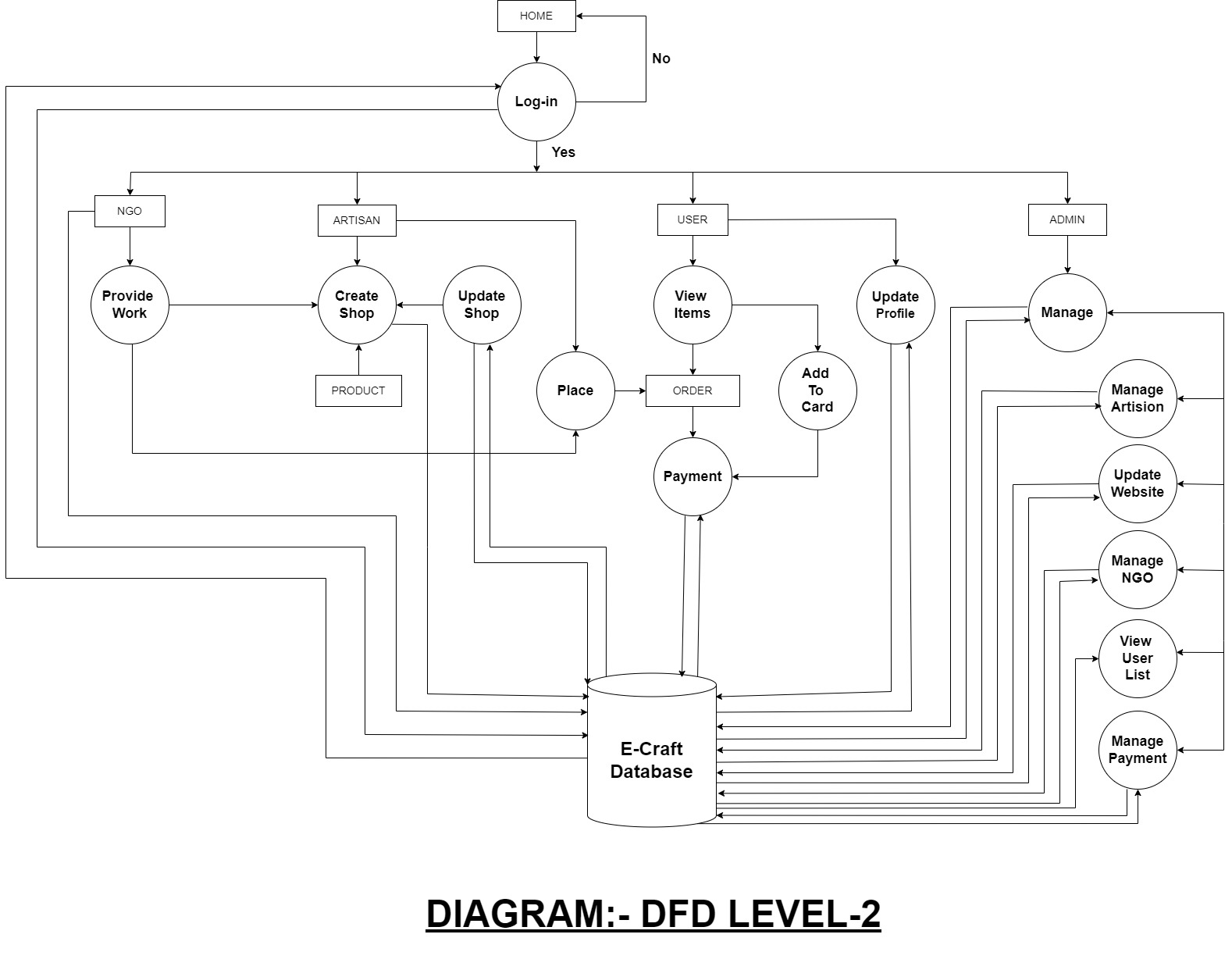
DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analysed or modelled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

**b. DFD Level 1: -**

 In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

**d. DFD Level-2: -**

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system’s functioning.

We have already seen how a level 0 context diagram can be decomposed (exploded) into a level 1 DFD. In DFD modelling terms we talk of the context diagram as the “parent” and the level 1 diagram as the “child”. This same process can be applied to each process appearing within a level 1 DFD. A DFD that represents a decomposed level 1 DFD process is called a level 2 DFD. There can be a level 2 DFD for each process that appears in the level 1 DFD.

### 

**CHAPTER-5**

**5.CODING**

**HTML OF LOGIN :-**

{% include "app/header.html" %}

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Sign-In</title>

<link rel="stylesheet" href="{% static " app/css/login.css" %}" />

<script src="//cdnjs.cloudflare.com/ajax/libs/annyang/2.6.1/annyang.min.js"></script>

<script>

if (annyang) {

var commands = {

'alert': () => { alert('Hello Sir welcome to Ecraft Website'); }

};

var khushal = {

'username \*tag': function (variable) {

let uname = document.getElementById("uname");

uname.value = variable;

}

}

var khushall = {

'password \*tag': function (variable) {

let pas = document.getElementById("pas");

pas.value = variable;

}

}

// Add our commands to annyang

annyang.addCommands(commands);

annyang.addCommands(khushal);

annyang.addCommands(khushall);

// Start listening.

annyang.start();

}

</script>

</head>

<body>

<section class="log">

<div class="container1">

<div class="forms-container">

<div class="left-sign">

<div class="containt">

<img src="{% static " app/images/logo.png" %}" alt="" class="logo2">

<h1>Welcome Back </h1>

<p>

to get started please login with your personal info

</p>

</div>

</div>

<div class="signin">

<form id="fr2" method="POST" action="" class="sign-in-form">

{% csrf\_token %}

<h2 class="title">Sign In</h2>

<div class="input-field1">

<img src="{% static " app/images/user\_img-removebg-preview.png" %}" alt="" height="75%" width="55%"

style="padding-top: 0.8rem; padding-left: 0.2rem;">

<input name="username" required id="uname" type="text" placeholder="Username" class="input1" />

</div>

<div class="input-field1">

<img src="{% static " app/images/Passwordimg.png" %}" alt="" height="75%" width="55%"

style="padding-top: 0.8rem; padding-left: 0.2rem;">

<input type="password" id="pas" required name="password" placeholder="Password" class="input3" />

</div>

<div class="btn-box">

<button type="submit" id="but" name="submit" class="btn solid">Sign In</button>

</div>

<a href="{% url 'reset\_password'%}" class="btn1" style="text-decoration:none; color:black;">Forgot

Password</a>

<h5 class="h5">Don't have an Account? <a href="/register" style="text-decoration:none;">SignUp</a></h5>

</div>

</form>

</div>

</div>

</div>

</section>

{% include "app/footer.html" %}

</body>

</html>

**HTML OF SIGNUP :-**

{% load static %}

{% include "app/header.html" %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Sign-Up</title>

<link rel="stylesheet" href="{% static " app/css/signup.css" %}" />

</head>

<body>

<section class="log">

<div class="container1">

<div class="forms-container">

<div class="left-sign">

<div class="containt">

<img src="{% static " app/images/logo.png" %}" alt="" class="logo2">

<h1>Looks like you're <br> new here!</h1> <br>

<p>

Sign up with your personal information <br> to get started

</p>

</div>

</div>

<div class="signin">

<form id="fr1" method="POST" action="" class="sign-in-form">

{% if messages %}

{% for message in messages %}

<p {% if message.tags %} class="alert alert-{{message.tags}}" {% endif %}>{{message}}</p>

{% endfor %}

{% endif %}

{% csrf\_token %}

<h2 class="title1">Sign Up</h2>

<div class="input-field1">

<img src="{% static " app/images/user\_img-removebg-preview.png" %}" alt="" height="75%" width="55%"

style="padding-top: 0.8rem; padding-left: 0.2rem;">

<input name="username" required type="text" placeholder="Username" class="input1" />

</div>

<div class="input-field1">

<img src="{% static " app/images/Emailimg.png" %}" alt="" height="75%" width="55%"

style="padding-top: 0.8rem; padding-left: 0.2rem;">

<input type="email" required name="email" placeholder="Email ID" class="input2" />

</div>

<div class="input-field1">

<img src="{% static " app/images/Passwordimg.png" %}" alt="" height="75%" width="55%"

style="padding-top: 0.8rem; padding-left: 0.2rem;">

<input type="password" required name="password1" placeholder="Password" class="input3" />

</div>

<div class="input-field1">

<img src="{% static " app/images/Passwordimg.png" %}" alt="" height="75%" width="55%"

style="padding-top: 0.8rem; padding-left: 0.2rem;">

<input type="password" required name="password2" placeholder="conform Password" class="input3" />

</div>

<div class="btn-box">

<button type="submit" id="but" name="submit" class="btn solid">Sign Up</button>

</div>

<h5 class="h5">

Alredy have an Account? <a href="/login" style="text-decoration:none;">Login </a></h5>

</div>

</form>

</div>

</div>

</div>

</section>

{% include "app/footer.html" %}

</body>

</html>

CSS OF SIGNUP :-

@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800&display=swap");

\* {

margin: 0px;

padding: 0px;

box-sizing: border-box;

cursor: pointer;

}

html body {

height: 100%;

width: 100%;

}

body,

input {

font-family: "Poppins", sans-serif;

background: #f9eeee;

}

.left-sign {

background:#f3ab00;

color: #ffffff;

height: 90%;

padding: 0 2rem;

width: auto;

display: flex;

justify-content: center;

align-items: center;

text-align: center;

>.containt {

padding: 2rem 0;

>h1 {

line-height: 2rem;

>p {

line-height: 1.5rem;

}

}

}

.log{

width: 100%;

height: 100%;

}

.logo2 {

height: 15%;

width: 40%;

}

.container1 {

display: flex;

justify-content: space-around;

flex-direction: column;

width: 100%;

background-color: #ffffff;

}

.forms-container {

position: absolute;

width: 100%;

height: 95%;

top: 20%;

left: 0;

display: flex;

justify-content: center;

align-items: center;

}

.signin {

display: grid;

grid-template-columns: 1fr;

z-index: 5;

background: #ffffff;

width: 30%;

height: 90%;

padding: 26px 46px;

}

#fr1{

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

transition: all 0.2s 0.7s;

overflow: hidden;

grid-column: 1 / 2;

grid-row: 1 / 2;

}

form.sign-in-form {

z-index: 2;

}

.title2 {

font-size: 2.2rem;

color: black;

margin-bottom: 10px;

padding: 0px 106px;

}

.input-field1 {

max-width: 380px;

width: 100%;

background-color: #f0f0f0;

margin: 10px 0;

height: 55px;

border-radius: 55px;

display: grid;

grid-template-columns: 15% 85%;

padding: 0 0.4rem;

position: relative;

}

.input-field1 .input1 {

background: unset;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;

border-radius: 30px;

}

.input-field1 .input2 {

background: unset;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;

border-radius: 30px;

}

.input-field1 .input3 {

background: unset;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;

border-radius: 30px;

}

.input-field1 input::placeholder {

color: grey;

font-weight: 500;

}

.social-text {

padding: 0.7rem 0;

font-size: 1rem;

}

.social-media {

display: flex;

justify-content: center;

}

.social-icon {

height: 46px;

width: 46px;

display: flex;

justify-content: center;

align-items: center;

margin: 0 0.45rem;

color: #333;

border-radius: 50%;

border: 1px solid #333;

text-decoration: none;

font-size: 1.1rem;

transition: 0.3s;

}

.social-media a i {

color: #333;

}

.social-icon:hover {

color: #4481eb;

border-color: #4481eb;

}

.btn-box .btn{

display: inline-flex ;

justify-content: center;

align-items: center;

transition: .6s;

overflow: hidden;

position: relative;

z-index: 1;

width: 150px;

background-color: transparent;

border: 2px solid green;

height: 49px;

border-radius: 49px;

color: #000;

text-transform: uppercase;

font-weight: 600;

margin: 12px 115px;

}

.btn:hover{

color: #000;

}

.btn::before{

content: "";

top: 0;

left: 0;

position: absolute;

width: 0;

height: 100%;

transition: .5s;

background:transparent;

z-index: -1;

}

.btn:hover::before{

width: 100%;

}

.btn:hover{

color: #ffffff;

border: none;

box-shadow: #f3ab00 0px 3px 7px;

}

.btn::before{

background:#f3ab00;

}

.btn1{

padding-top: 1rem;

color: #000;

}

.btn1:hover{

text-decoration: dashed;

color: green;

}

.h5{

margin-top: 3rem;

>a:hover{

color: green;

}

}

CSS OF SIGNIN :-

@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800&display=swap");

\* {

margin: 0px;

padding: 0px;

box-sizing: border-box;

cursor: pointer;

}

html body {

height: 100%;

width: 100%;

}

body,input {

font-family: "Poppins", sans-serif;

background: #f9eeee;

}

.left-sign {

background:#f3ab00;

color: #ffffff;

height: 90%;

padding: 0 2rem;

width: auto;

display: flex;

justify-content: center;

align-items: center;

text-align: center;

>.containt {

padding: 2rem 0;

>h1 {

line-height: 1rem;

}

>p {

line-height: 4rem;

}

}

}

.log{

width: 100%;

height: 100%;

}

.logo2 {

height: 15%;

width: 40%;

}

.container1 {

display: flex;

justify-content: space-around;

flex-direction: column;

width: 100%;

background-color: #ffffff;

}

.forms-container {

position: absolute;

width: 100%;

height: 95%;

top: 20%;

left: 0;

display: flex;

justify-content: center;

align-items: center;

}

.signin {

display: grid;

grid-template-columns: 1fr;

z-index: 5;

background: #ffffff;

width: 30%;

height: 90%;

padding: 26px 46px;

}

#fr2{

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

transition: all 0.2s 0.7s;

overflow: hidden;

grid-column: 1 / 2;

grid-row: 1 / 2;

}

form.sign-in-form {

z-index: 2;

}

.title2 {

font-size: 2.2rem;

color: black;

margin-bottom: 10px;

padding: 0px 126px;

}

.input-field1 {

max-width: 380px;

width: 100%;

background-color: #f0f0f0;

margin: 10px 0;

height: 55px;

border-radius: 55px;

display: grid;

grid-template-columns: 15% 85%;

padding: 0 0.4rem;

position: relative;

}

.input-field1 .input1 {

background: unset;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;

border-radius: 30px;

}

.input-field1 .input2 {

background: unset;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;

border-radius: 30px;

}

.input-field1 .input3 {

background: unset;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;

border-radius: 30px;

}

.input-field1 input::placeholder {

color: grey;

font-weight: 500;

}

.social-text {

padding: 0.7rem 0;

font-size: 1rem;

}

.social-media {

display: flex;

justify-content: center;

}

.social-icon {

height: 46px;

width: 46px;

display: flex;

justify-content: center;

align-items: center;

margin: 0 0.45rem;

color: #333;

border-radius: 50%;

border: 1px solid #333;

text-decoration: none;

font-size: 1.1rem;

transition: 0.3s;

}

.social-media a i {

color: #333;

}

.social-icon:hover {

color: #4481eb;

border-color: #4481eb;

}

.btn-box .btn {

display: inline-flex;

justify-content: center;

align-items: center;

transition: .6s;

overflow: hidden;

position: relative;

z-index: 1;

width: 150px;

border: 2px solid rgb(4, 209, 4);

background-color: rgb(4, 209, 4);

height: 50px;

border-radius: 49px;

color: #000;

text-transform: uppercase;

font-weight: 600;

margin: 12px 115px;

}

.btn::before {

content: "";

top: 0;

left: 0;

position: absolute;

width: 0;

height: 100%;

transition: .5s;

background: transparent;

z-index: -1;

}

.btn:hover::before {

width: 100%;

}

.btn:hover {

color: #ffffff;

border: none;

box-shadow: #f3ab00 0px 3px 7px;

}

.btn::before {

background: #f3ab00;

}

.btn3{

padding-top: 1rem;

color: #000;

text-decoration: none;

margin: 1px 105px;

}

.btn3:hover{

text-decoration: dashed;

color: green;

}

.h5{

margin-top: 7rem;

>a:hover{

color: green;

}

}

**CHAPTER-6**

### **6.TESTING**

**What is Testing: -**

Software testing is a process whose purpose is to evaluate the functionality of software application and to find out whether the developed software meets the specified requirements and does not contain any errors.

**Why Need Testing: -**

While making software, we make some mistakes are harmful, then some human beings make mistakes all the time. Some mistakes are not harmful, then some mistakes can be very dangerous. To rectify all these mistakes, we have to test the software.

1. Software testing is done to find all possible defects and errors in the software.
2. Software testing is also done to ensure the quality of the software product. If the quality is better than customers will use a greater number of software products.
3. To increase the performance of the software.
4. To prove that there is no fault in the software.
5. To ensure that the software is made as per the requirement of the customer or not?
6. Software testing is also done to stay in business.

**Type of Testing: -**

MANUAL TESTING

WHITE BOX TESTING BLACH BOX TESTING GRAY BOX TESTING

**A. WHITE BOX TESTING: -**

White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass called glass testing or open box testing. In order to perform white box testing on an application a tester needs to know internal work

Input

output

##### **B. BLACK BOX TESTING: -**

The technique of testing without having any knowledge of the interior working of the application is called black box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black box text, a tester will interact with the system user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.



BLACK BOX TESTING

INPUT OUTPUT

#### **C. GRAY BOX TESTING:** -

Gray Box Testing is a software testing technique which is a combination of Black Box Testing technique and White Box Testing technique. In Black Box Testing technique, tester is unknown to the internal structure of the item being tested and in White Box White Box Testing the internal structure is known to tester. The internal structure is partially known in Gray Box Testing.

|  |
| --- |
| Gray Box Testing |

|  |
| --- |
| White Box Testing |

|  |
| --- |
| Testing  Black Box |

## + =

**FUNCTIONAL TESTING: -**

### **1.UNIT TESTING: -**

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units. It is often done by programmer by using sample input observing its corresponding outputs.

### **2.** **INTEGRATION TESTING: -**

Integration testing is defined as the testing of combine parts of application to determine if function correctly.

Integration testing is of four types: -

* Top Down
* Bottom Up
* Sandwich
* Big Bang

#### **Bottom-up integration: -**

In this testing begins with unit testing, followed by tests of progressively higher-level module is thereafter.

#### **Top-down integration: -**

In this testing higher level modules are tested first and progressively, lower-level module is thereafter.

### **3. SYSTEM TESTING: -**

In this software is tested such that it works fine for different operating system. It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working. In this we have security testing, street, recovery testing and performance testing.

### **4. ACCEPTANCE TESTING: -**

Acceptance testing is a level of software testing where a system is tested for acceptability. Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system’s compliance with the business requirements and verify if it is having met the required criteria for delivery to end users.

There are various forms of acceptance testing:

1. User Acceptance
2. Testing Business
3. Alpha Testing
4. Beta Testing

### **1. ALPHA TESTING: -**

Alpha testing is a type of acceptance testing, performed to identify all possible issues/ bugs before releasing the product to everyday users or the public. Alpha Testing performed at developer’s site. Reliability and Security Testing are not performed in depth Alpha Testing. Alpha testing involves both the white box and black box techniques.

### **2. BETA TESTING: -**

Beta Testing of a product is performed by “real users” of the software application in a

“Real environment” and can be considered as a form of external User Acceptance Testing. Beta testing is performed by clients or End Users who are not employees of the organization. Beta Testing typically uses Black box Resting.

Unit

Testing

Integration

Testing

Beta

Testing

Alpha

Testing

System

Testing

**NON-FUNCTIONAL TESTING: -**

### **1. PERFORMANCE TESTING: -**

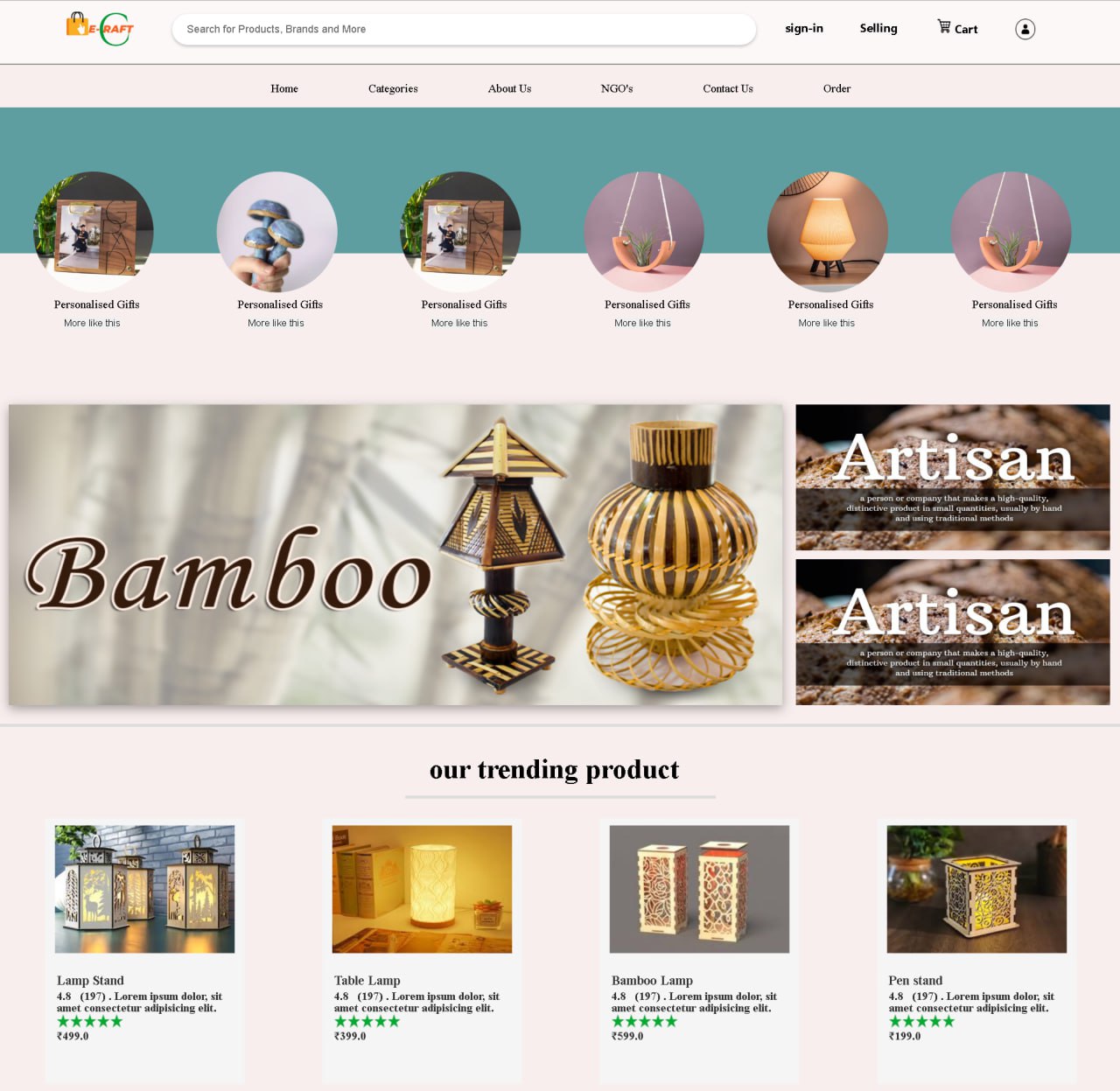
Performance testing a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measure the quality attributes of the system, such as scalability, reliability and resource usage.

Performance Testing Techniques: -

* **Load testing-** It is the simplest form of testing conducted to understand behaviour of the system under a specific load. load testing will result in measuring import business critical transactions and load on the database, application server etc. are also monitored.
* **Stress testing-** It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above expected maximum.

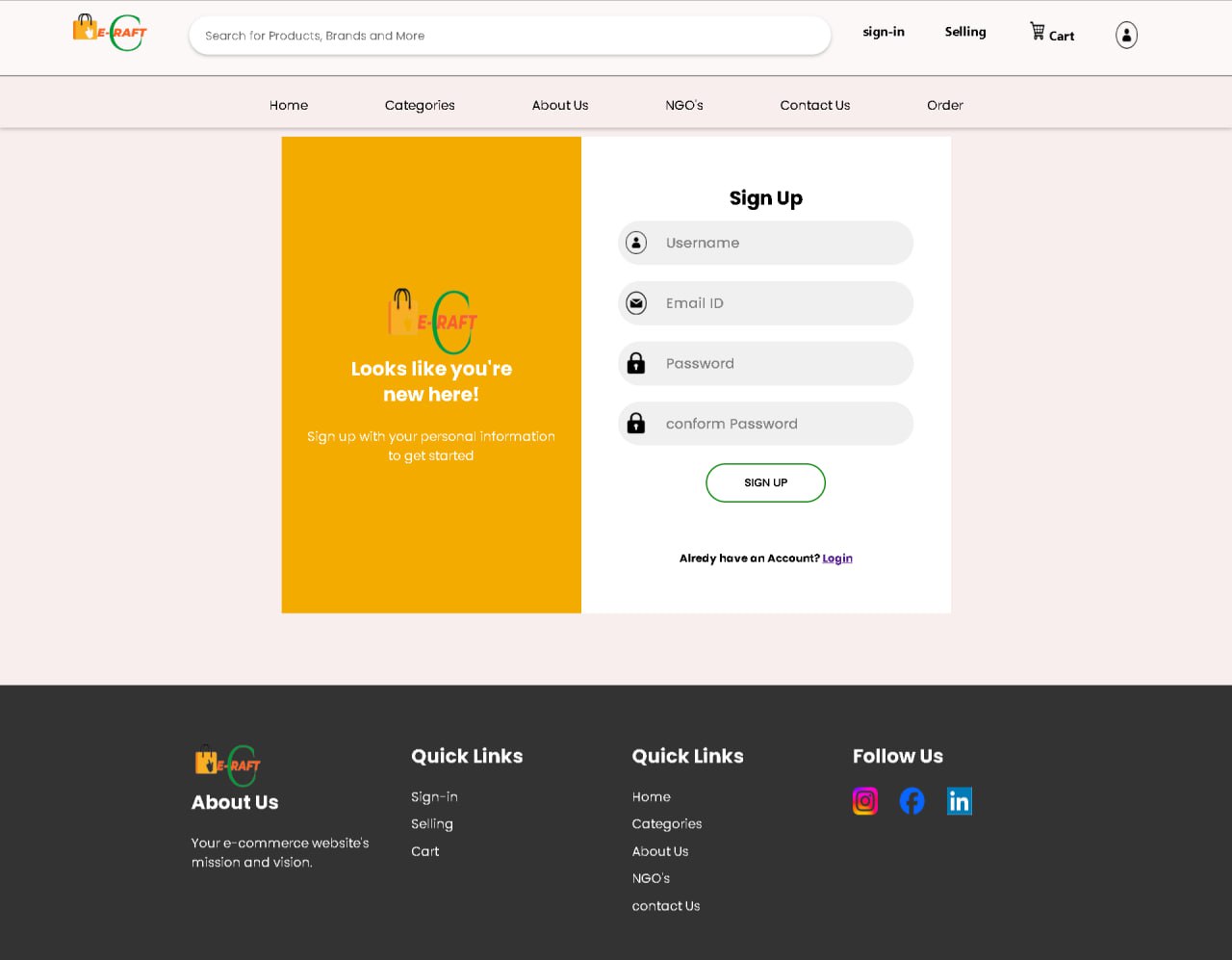
**CHAPTER–7**

### 

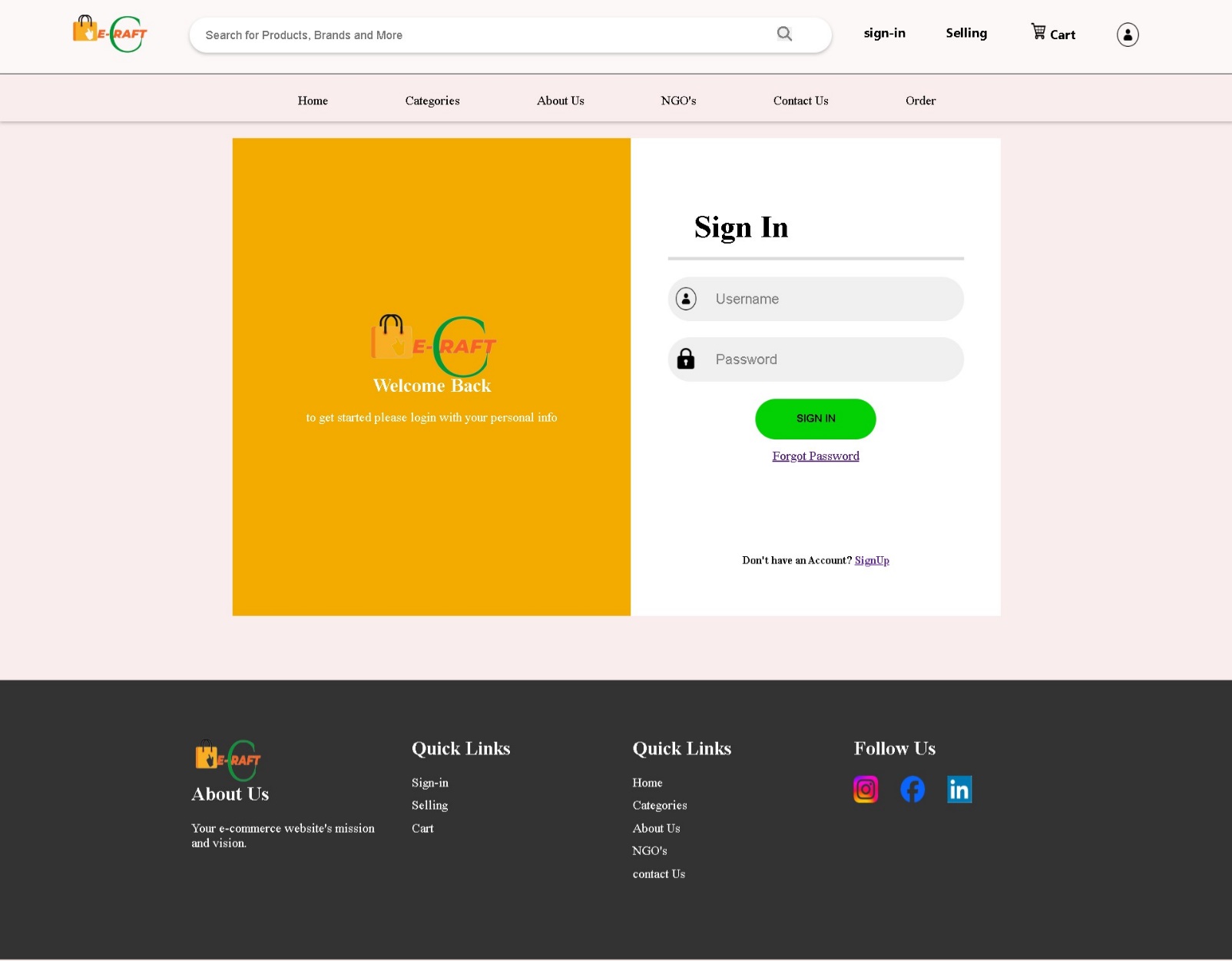
**7.OUTPUT SCREEN** 

**7.1 Home Page (Before Login): -**

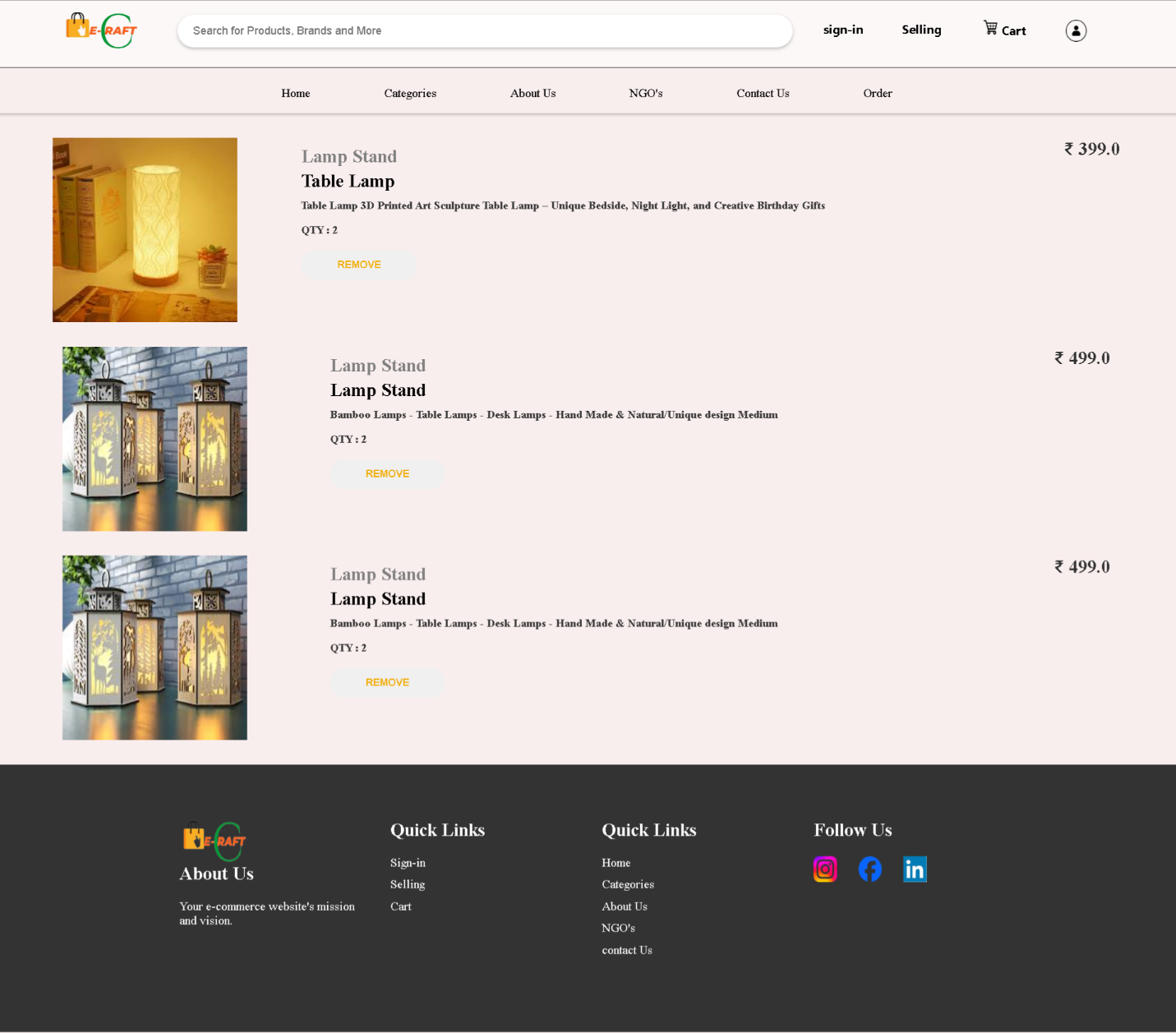
**7.2 Sign Up Page: -**



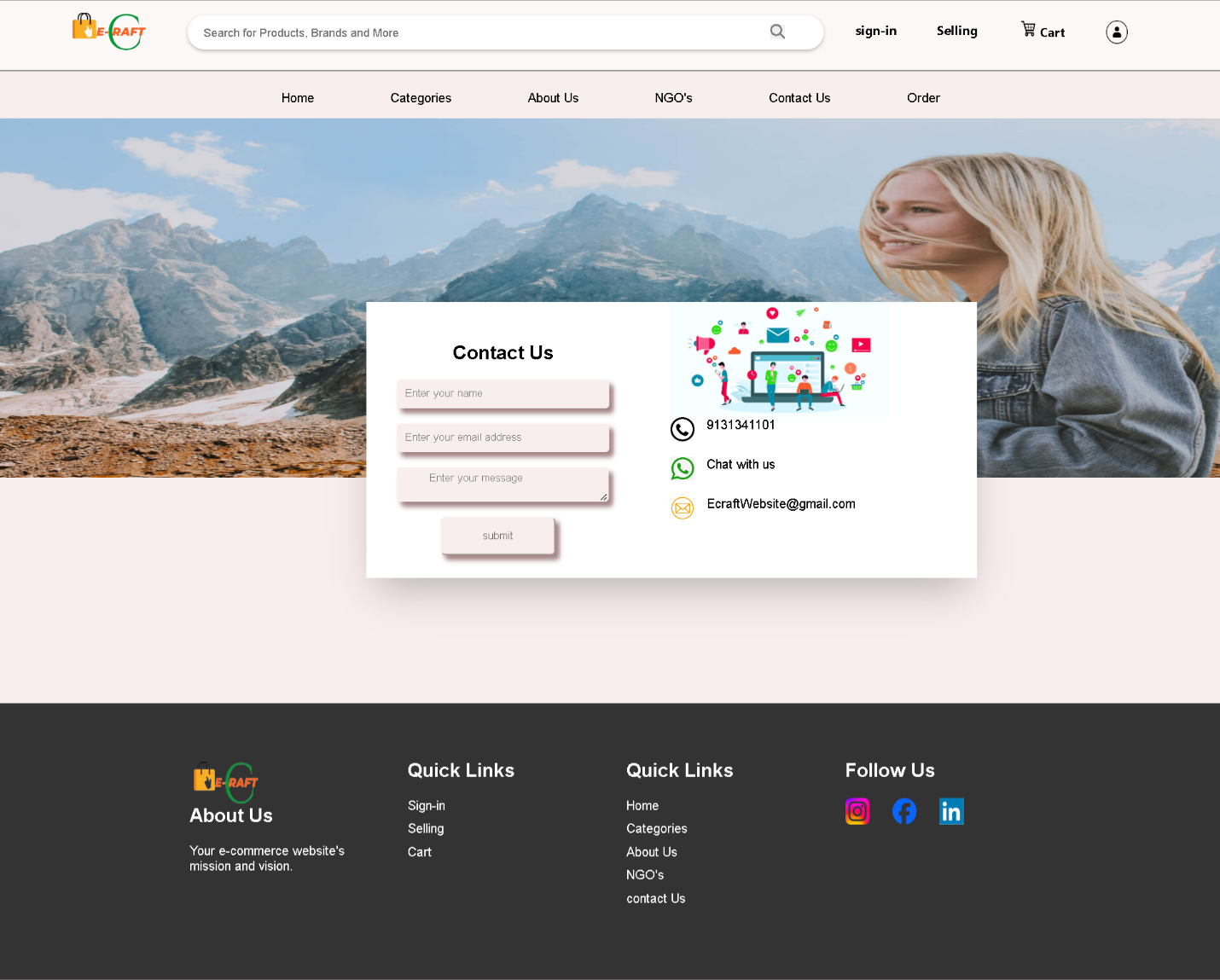
**7.3 Sign-In: -**



**7.3 Search : -**

****

**7.5Contact us : -**

****

**CHAPTER-8**

**8.COST ESTIMATION**

The constructive cost model (COCOMO) is a procedural software cost estimation model development by Barry W. Boehm. The model parameters are derived from fitting a registration formula using data from historical project (61 project for COCOMO 81 and 63 projects for COCOMO 1) Basic COCOMO computer software development effort (and cost) as a function of program size is expressed in estimate thousands of sources line of code (SDLC, KLOC). COCOMO applies to three classes of software project:

1. **Organic projects** – “Small” teams with “good” experience working with “less then rigid” requirement.
2. **Semi-detached projects** – “Medium” teams with mixed experience working with “less then rigid” requirement.

#### **3. Embedded project –**

Developed within a set of “tight” constrains. It is also combination of organic and semi-detached projects (Hardware, software, operational). The estimated effort and scheduled time for the project are given by the relation:

Effort (E) = a\*(KLOC) ^b PM

**Scheduled time (D) = c\*(E) ^d M {**M: months}

**Average resource size (Required Persons) = E/D P {P: Persons}**

**Productivity of Software = KLOC/E**

Where,

E = Total effort required for the project in Persons-Months (PM).

D = Total time required for the project Development in months (M).

KLOC = The size of the code for the project in kilo lines of code.

a, b, c, d = The constants parameters for a software project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODE** | **a** | **B** | **c** | **d** |
| **Organic** | 2.4 | 1.05 | 2.5 | 0.38 |
| **Semi-detached** | 3.0 | 1.12 | 2.5 | 0.35 |
| **Embedded** | 3.6 | 1.20 | 2.5 | 0.32 |

**CALCULATION: -**

**Effort (E): -**

E = a(KLOC)^b

E = 2.4 \*(6.5) ^1.05

E = 2.4 \* 7.1377

E = 17.14 **Person-Months**

**Development Time: -**

DT = C\*(E)^d

DT = 2.5 \*(17.14) ^0.38

DT = 7.3 **Months**

**Average Staff Size: -**

ASS = Effect / Dev. Time

ASS = 17.14 / 7.3

ASS = 2.34 **Persons**

**Productivity: -**

P = KLOC/ Effect

P = 6.5 / 17.14

P = 0.379 **KLOC/P.M.**

**Total Salary: -**

**1 Person** **= 9000**

**Then,**

**Total salary =** Total persons \*1 Person Salary

**Total salary =** 7 \* 9000

**Total salary = 63000 Rs.**

**CHAPTER-9**

**9. CONCLUSION**

The main objective of E-Craft Commercial Website is to enhance the art of local area artisan and they have to provide work through this platform. This platform is a simple and easy for their work and to fill their arts. The objective of the craft website is to make various artisan of the local or area and their arts accessible to the people through the website. This website will have many crafts like bamboo craft, metal craft, jute craft etc.

Which the user can purchase and sell. Through this website, the artisan can work within the NGO and the user can also create his own shop and sell the product there. Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to use his skill so that he can benefit from it.

In the platform, work will be provided to the users among NGOs. Through this website, the problem between the artisan and the NGO will be solved, such as the artisan does not know where to practice his skills so that he can benefit from it. In this craft, work will be provided to the users through the NGO.

**CHAPTER-10**

**BIBLIOGRAPHY**

1. https://ehindistudy.com
2. https://www.javatpoint.com/uml-use-case-diagram
3. https://www.geeksforgeeks.org/software-engineering-cocomo-model/
4. <https://www.w3schools.com>
5. https://www.amazon.com
6. https://www.flipkart.com
7. <https://medium.com/@mahmoud.nasser.abdulhamed11/best-django-books-1c1ea309ee35>
8. Book: - HTML and CSS: Design and Build Websites

Writer: - Jon Duckett

1. Book: - Django 2 Web Development Cookbook

Writer: - Aidas Bendoraitis and Jake Kronika

1. Book: - Clean Code: A Handbook of Agile Software Craftsmanship

Writer: - Robert C. Martin

1. Some NGO’S Name: -

* Adiwasi swayam kala sansthan(ASKS)

Address :- Uikey Shilpgram Salekasa po.&tah. :- Salekasa , Dist. :- Gondia MH , 441916

* Neware Handicraft

Address :- Wrd No. :-32 , Moti Nagar , Balaghat(M.P.) , 481001

1. Some Artisan Name: -

* Rakesh Neware 9685658408
* Usha Sahare 8458838934
* Bharti 8085947408
* Sudheer Neware 8817051131