CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Augmented reality is a technology that superimposed virtual object into real world. The jewellery store application is the Augmented reality-based application in which user can wear the jewellery ornaments in real time. The App allows users to experience a visual representation of ornaments through their web application .The use of AR would allow customers to view their jewellery on their fingers and neck to gain a better idea of how it would look once completed.

Here in this system the ICP algorithm is used for the 3D projection of the object for the inspection of the user before trial. The program takes the input as an image of the customer who is standing in front of screen which provides access to customer to see how they look in it. Thus, the system will help to customer for taking the decision of buying the ornament. This is achieved by using webcam. System will detect and track the movement of the customer as soon as the customer selects a particular item to from application that particular item will adjust itself onto that relevant face part of customer and hence the customer can try out various ornaments and decide whether that particular item suits him/her before ordering them.

1.2 GOAL OF THE PROJECT

It bridges the gap between virtual reality and the real world, and popularly used in visual art.AR has the potential of assisting shoppers in creating jewellery, and it provides a virtual picture of how well it looks on them before authorizing the purchase. The in-store jewellery sells can be enhanced through the use of AR equipped screens, mirrors and dressing rooms. The use of such solutions will attract and impress more jewellery shoppers. In this pandemic situation customer can buy ornaments through a web application with AR experience.

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF SIMILAR WORK

Home depot

The Home Depot, Inc., commonly known as Home Depot, is the largest home improvement retailer in the United States, supplying tools, construction products, and services. In 2017 Home Depot expanded the functionality in its main mobile app to allow users to overlay Home Depot merchandise on any room in their home.  The app can ascertain the dimensions of a select number of Home Depot items ranging from refrigerators to chandeliers. That enables the consumer to have a 3D visual of a product in any space they want to see a product in. This app is worth a try as Forrester Research recently ranked Home Depot’s mobile app as the number one retailer mobile app due to its functionality and user experience.

Nike Fit

Nike added a new augmented reality feature in its app called Nike Fit. The feature allows customers to find their true shoe size. To enable the app’s functionality all users have to do is point their phone’s camera at their feet and the app will determine their shoe size. All of this this can be done from the comfort of the customer’s home in less than a minute. The tool which is a part of the existing Nike app – uses augmented-reality technology to scan your feet with a smartphone camera, mapping each foot's dimensions using a 13-point measuring system Once the customer’s shoe size is determined it is saved in the app. If a customer shops in-store all a sales associate has to do is scan a QR code in the app and they can retrieve the customer’s shoe size. With more than an estimated 60% of people wearing the wrong size shoe size this app will help customers to get the shoe that fits.

2.1.1 EXISTING SYSTEM

The current system is selling ornaments through retailer shops only. There are some online shopping sites for ornaments but they were not provide any augmented reality experience to their customers. So the only option is to buy ornaments is from retailer shop, then only a customer can have a real time experience. The online shopping sites only display the products that cannot determine how that will be in our body parts.

2.1.2 DRAWBACKS OF EXISTING SYSTEM

When we buy ornaments from a jewellery shop we try trail before buy it. The ornaments are precious, that is why it’s important to take a good care of them. Customers used to handles the ornaments at the time of shopping, because of which ornaments can getting harmed or getting dirty and if they are not handle carefully might also be chances of breaking down.

Buying ornaments through online shopping sites is not preferred by customers, because people only buy ornaments through a real time experience. Through an ecommerce site customers can’t experience the jewelry, without that no one want to buy expensive jewelry items from ecommerce sites.

CHAPTER 3

OVERALL DESCRIPTION

3.1 PROPOSED SYSTEM

Implementing augmented reality to ecommerce site of a jewelry shop. The program takes the input as an image of the customer who is standing in front of screen which provides access to customer to see how they look in it. Thus, the system will help to customer for taking the decision of buying the ornament. The augmented reality application lets the customer explore the finest jewellery experience by projecting the real-looking impressions. This will benefit not only the customer but also the seller. As the retailer doesn’t need to show any piece physically to the buyer. Even in this pandemic situation it’s better to buy ornaments through online with a real time experience with the help of augmented reality.

3.2 FEATURES OF PROPOSED SYSTEM

* Detection of neck and hand- Using ICP algorithm, system identify customer’s neck and hand using a camera in front of them and place ornaments to their desired part.
* Simulation of real time ornament trail- Customers can experience a real time ornament trail from where ever they, by deciding it using augmented reality it give a new buying experience for customers.
* View products- View every ornaments in this web application and can experience how it will be in our hand and neck.
* Suggest suitable ornaments- By detecting the body parts and identify which size is suitable for the customer and suggest the items for them. Also analyze their ornament search results and predict the desires of the customer.

3.3 FUNCTIONS OF PROPOSED SYSTEM

* Identifying a person from the video and detect their neck and hand.
* Measure ring size automatically to reduce the risk of a poor fit.
* Browse and make customers choice with the virtual try-on experience
* View product catalogue based on customer preference
* Take trial photos and search for product based on this picture
* Building an object virtually 3D to use in a jewellery shop in which a virtual object will be placed instead of actual ornament in a real world using augmented reality.
* System will detect and track the movement of the customer as soon as the customer selects a particular item to from application, that particular item will adjust itself onto that relevant face part of customer and hence the customer can try out various ornaments and decide whether that particular item suits him/her or not by them experience like manual shopping.

3.4 REQUIREMENTS SPECIFICATION

System analyst tasks to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaires, details investigation, observation, collection of samples etc. As the details are collected, the analyst study the requirements data to identify the features the new system should have, including both the information the system produce and operational features such as processing controls, response times, and input output methods.

Requirement specification simply means, “Figuring out what to make before you make it”. It determines what people need before you start developing a product for them. Requirement definition is the activity of translating the information gathered in to a document that defines a set of requirements. These should accurately reflect what consumer wants. It is an abstract description of the services that the system should provide and the constraints under the system must operate.

The requirements fall into two categories: functional requirements and non-functional requirements.

The requirements of specification of the proposed system are as follows:

* Minimum time needed for various processing
* Better Service
* Faster response time

3.5 FEASIBILITY ANALYSIS

* The main aim of the feasibility study activity is to determine. Whether it would be financially and technically feasible to develop the product. The feasibility study activity involves analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system the processing required to be carried out of these data, the output data required to be carried out of these data, the output data required to be produced by the system, as well as various constraints on the behavior of the system.
* In our software we would find the actual requirements of this software and add that features Such as monitoring, process scanning etc. For adding this feature, we will like take different ways to solving this last find the best way to complete these features.
* Feasibility studies aim to objectively and rationally uncover the strengths and weakness of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or vice, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, studies precede technical development and project implementation.
  + 1. TECHNICAL FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project when writing a feasibility report, the following should be taken to consideration. A brief description of the business the part of the business being looked towards. The human and economic factor the possible solutions to the problems.

The system is technically feasible.

* + 1. OPERATIONAL FEASIBILITY

Operational analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and saving that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. Cost-based study: It is important to identify cost and benefit factors, which can be categorized as follows:

1. Development costs.
2. Operating costs.

This is an analysis of the costs to be incurred in the system and benefits derivable out of the system. Time-based study: This is an analysis of the time required to achieve a return on investments the future value of a project is also a factor. The system is operationally feasible.

* + 1. ECONOMICAL FEASIBILITY

In the economic feasibility the development cost of the system is evaluated weighing it against the ultimate benefit derived from the new system. Benefits can be tangible and intangible, direct or indirect. As assessment of the economic justification for a computer based system project is cost benefit. Cost benefit analysis is complicated by criteria that vary with the characteristics of the system to be developed, the relative size of the project and the respected return on investment derived as part of the strategic plan. The project AI Pedagogue is economically feasible because IDE used for developing software is free of cost.

3.5.4 BEHAVIORAL FEASIBILITY

Behavioral analysis is an operational principle for all requirements analysis methods. An estimate should be made of how strong a reaction the user is likely to have towards the development of a system. Behavioral analysis is an operational principle for all requirements analysis methods. The state-transition diagram represents the behavior of a system by depicting its status and the events that use the system to change state. AI Pedagogue is behaviorally feasible because of the effective use of the resource and also the system satisfies user needs and is user friendly.

CHAPTER 4

OPERATING ENVIRONMENT

4.1 HARDWARE REQUIREMENTS

Processor : Intel i5 8th Gen

RAM : 16GB ddr4

Hard disk : 2048 GB SSD

Drives : CD ROM, C-type Port, USB 3.1\*2 Port

Display Size : Compatible Size (Recommend 15’inch)

Screen Resolution : 1920\*1080 Pixels

Keyboard : Wireless Enabled Keyboard (Recommend: Logitech)

Keyboard Mouse : Wireless Enabled Mouse (Recommend: Logitech)

Monitor : Touch Capacity LED Monitor

Dedicated Graphics Card : Nvidia GeForce GTX 1050 4GB DDR5

Camera : 8 Megapixel Full HD 1.8f lens

Extra : Wi-Fi Adapter, Bluetooth Adapter

4.2 SOFTWARE REQUIREMENTS

Operating System : Windows (7/8/10)/Ubuntu (14/16/18/20)

Software Drivers : Wi-Fi drivers, Bluetooth Drivers, Visual Studio Drivers and Camera Drivers

Programming Language : Python

IDE : OpenCV, keras

Scripting Languages : HTML, CSS, JavaScript

Web Browser : Google Chrome

Front-End : Python, Django

Back-End : My SQL

4.3 Tools and platforms

4.3.1 PYTHON:

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. Python supports multiple programming pattern, including object oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.

**Features of Python**

* Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* Easy-to-read − Python code is more clearly defined and visible to the eyes.
* Easy-to-maintain − Python's source code is fairly easy-to-maintain.
* Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code. Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable − Can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* Databases − Python provides interfaces to all major commercial databases.
* GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* Scalable − Python provides a better structure and support for large programs than shell scripting.

4.3.2 MySql

MySQL Server is the world's most used relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. This stores data in the form of multiple related tables. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

**Features of MySQL**

* Relational Database Management System (RDBMS) - MySQL is a relational database management system.
* Easy to use - It is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.
* Secure - MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
* Free to download - MySQL is free to use and you can download it from MySQL official website.
* Scalable - MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.

CHAPTER 5

DESIGN

5.1 System Design

System can be defined, as an orderly grouping of interdependent components can be simple or complex. The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed .It refers to the technical specifications that will be applied in implementing the candidate system .It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the output and the inputs are also presented .In the second step, input data and master files are to be designed to meet requirement of the proposed output .The processing phase’s system’s objectives and complete documentation.

System design has two phases:

* Logical
* Physical

The logical design reviews the present physical system, prepares the input and output and also prepares a logical design walk- through .We have to deal with how to take entries required and whether and how to process the user data.

Physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation plan and new hardware and software. We have to decide how and where to store the input data and how to process it so as to present it to the user in an easy, informative and attractive manner.

Modules

* Admin
* User

About the project

Augmented Reality Based 3D Experience Jewellery Store is the software which gives customer an augmented reality experience in jewellery shopping through a web application. It records user’s videos and implement ornaments on them by detecting them. Below listed are some core features of our software but there are many functions and features provided in this software, within each of these are core modules.

* Admin

Administrative module controls the entire system, manage the entire application, core functions managed by admin are-

1. Product management

In product management function admin can add new product by entering details like its weight, product id, dimensions, features, price, quantity, tax and certificates. Admin can add categories and sort products upon its category. Pictures of the ornaments can be change in any time. Also admin can view the products and can edit the product details like weight, dimensions, certificates etc. Can block and unblock customer.

1. Order management

In order management function admin can view number of orders, order id and date of order, by using order id admin can view product id, product name, total quantity, amount, tax and discount of that product. Admin can also update and view the status of a particular product, view order report and customer feedback.

1. Inventory

Admin can add vendor by entering vendor id, total number of products, amount of tax and quantity. Also admin can view vendor details, edit vendor details, updating the number stock in the inventory and notify if stock gets empty or number of quantity is less than the limit. Enable or disable products and can add or delete vendor.

1. Coupon

In coupon management function admin can add coupon for additional discounts for particular products. User can have more discounts by using coupon codes. It is only distributed by the admin.

* User

In this module customer can view all the products and can search the product based on category and name. Customer can view the page with detailed description. By clicking the ornaments customer can experience a virtual trail facility in this application. Trails for bangles and necklaces are implemented in first phase.

Find the human posing in front of camera using OpenCV. Find the facial landmarks to obtain the 3D orientation of a human head with respect to the camera. For this pre trained human pose data set is used. The model takes as input a color image of size w × h and produces, as output, and locates the 2D locations of key points for each person in the image. The detection takes place in two stages:

**Stage 0**: The first 10 layers of the VGGNet are used to create feature maps for the input image.

**Stage 1**: A 2-branch multistage CNN is used where the first branch predicts a set of 2D confidence maps (S) of body part locations.

The most crucial part in the whole process of virtual trial is the control point for the placement of Jewellery.

The boundary points are segregated as two shoulders. This segregation is easily identifiable as they form two disjoint sets always. Each shoulder points are given to

Line Fitting algorithm by Least Square Method .The obtained control point has to be tracked in the subsequent frames so that whole process is not repeated again. If this control point is missed in any frame, then we need to re-calculate its position. Once the control point is obtained, we need to render the jewellery by suitably placing its end point with the control point. The end point of jewellery and its scaling factors are retrieved from the Metadata. The jewellery is scaled according to its scaling factor, and then jewellery image is superimposed over the user’s neck by replacing the pixels of jewellery with user’s body pixels

5.2 Data Flow Diagram

Data Flow Diagram (DFD) are directed graphs in which the nodes specify processing activities and the arcs that specify data items transmitted between processing nodes. Like flow charts DFD can be used at any desired level of abstraction. A DFD might represent data flow between individual statements or block of statements in a routine, data flow between concurrent process and data flow in a distributed computing system. Unlike flow charts DFD do not indicate a decision logic or condition under where various processing nodes in the diagram might be activated.

DFD is necessary for communicating for customer during requirement analysis; they are also widely used for representing external and internal design specifications. In the lack of structure DFD’s are quite valuable for establishing meaning, conventions and names of system components such as subsystems, files and data links. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently to make the data access easy, inexpensive and flexible to the user

A DFD consists of a series of bubbles joined by lines. The bubble represents data transformation and line represents data flow in the system. In the normal convention a DFD has four major symbols:

* Square, this defines source or destination of data.

|  |
| --- |
|  |

* Arrow, which shows data flow
* Circle, which represents a process that transforms incoming data into outgoing

flow.

* Open rectangle, which shows a data set.

## Figure 5.1 DFD Components

5.2.2 Project DFD

Level 0(CONTEXT LEVEL)

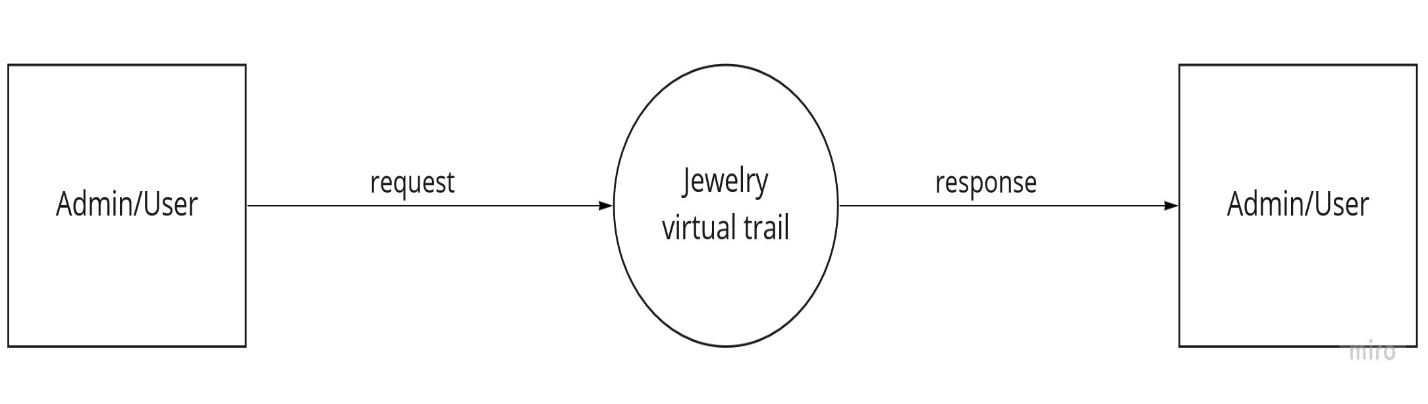


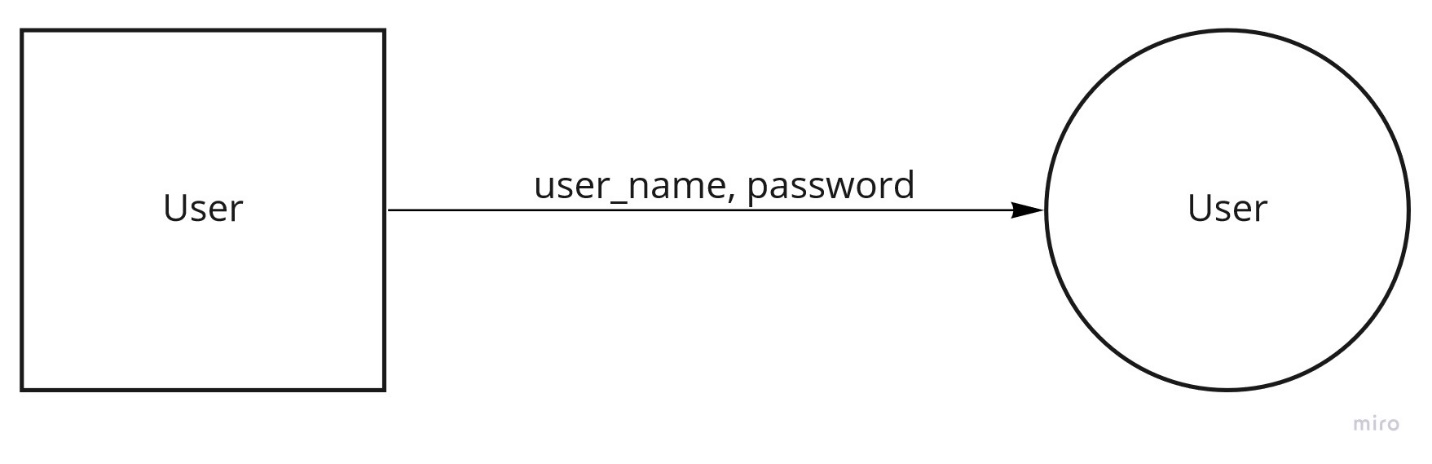
Figure 5.2 Context diagram (level 0)

Level 1 Admin login



Figure 5.3 Admin login (level 1)

Level 1 User login

 Figure 5.4 User login (level 1)

Level 1.0 Admin

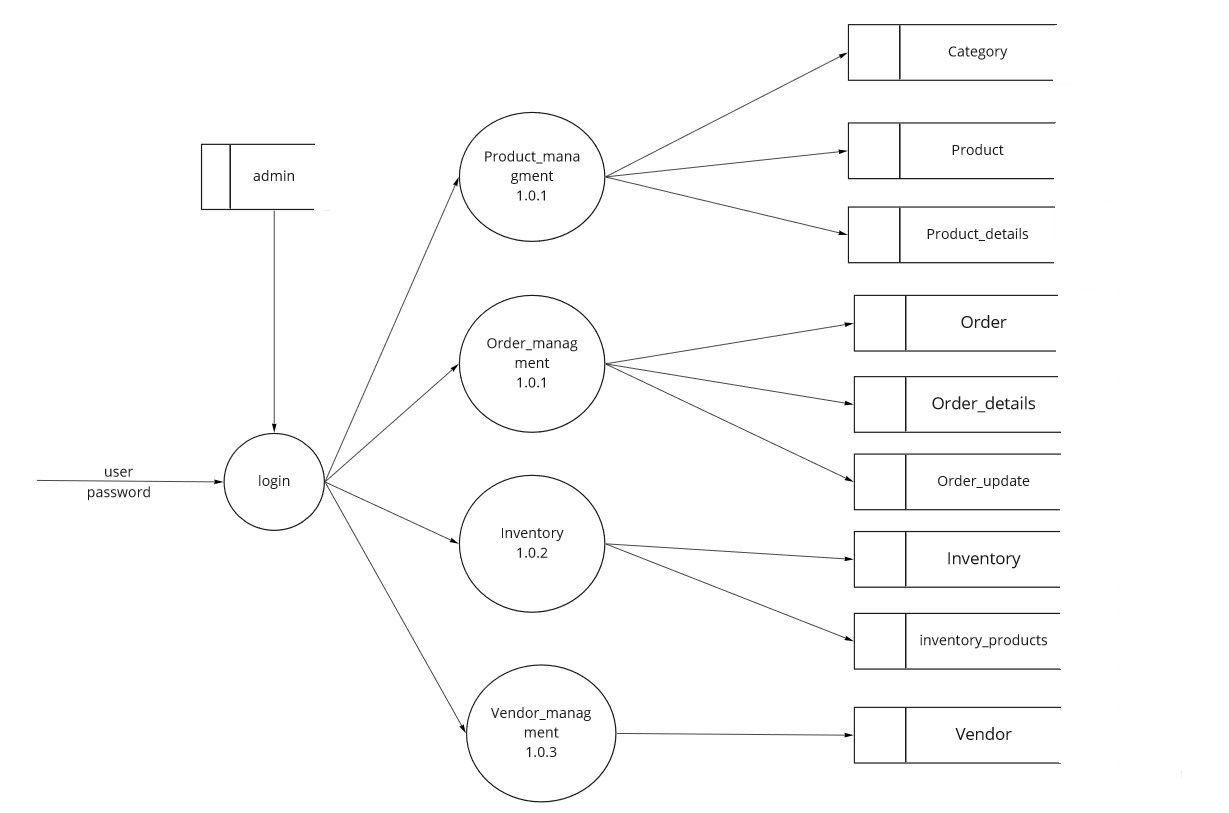


Figure 5.5 Admin (level 1.0)

Level 1.0 Admin (Product\_managment)

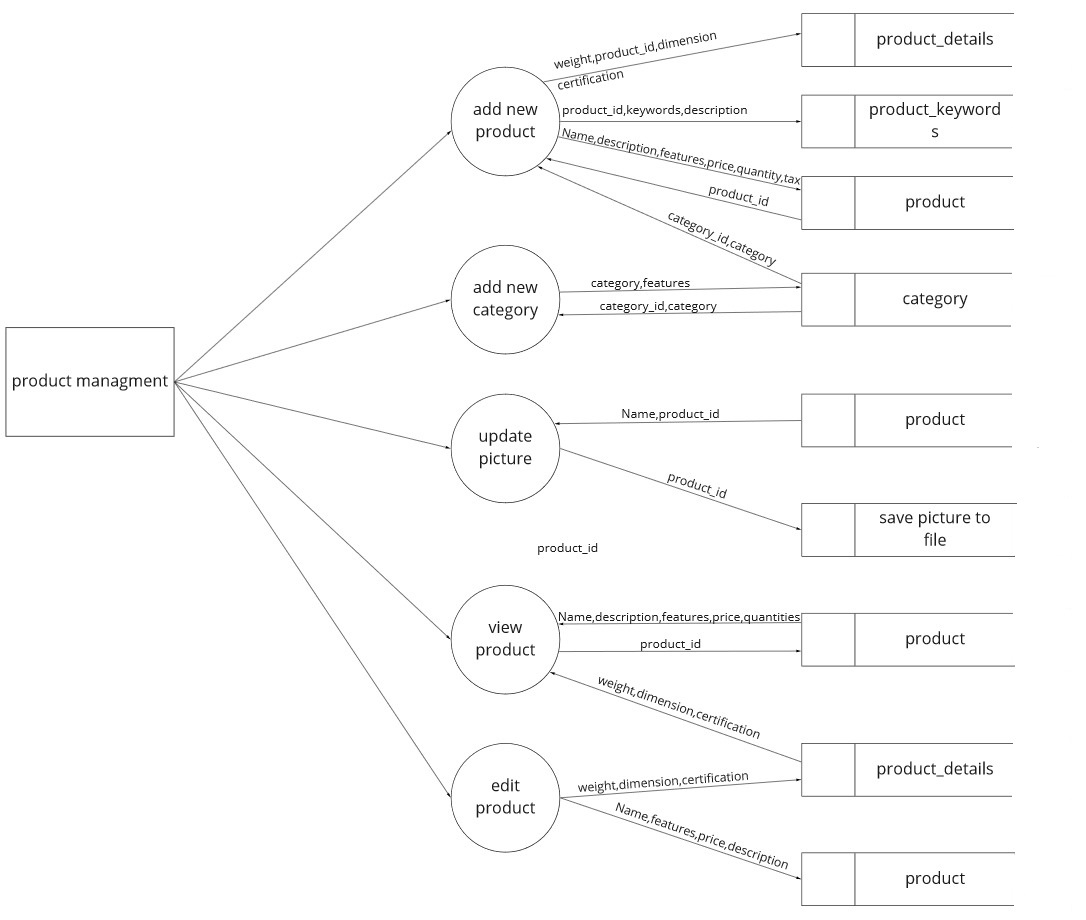


Figure 5.6 Product Management (level 1.0)

Level 1.0 Admin (Order\_managment)

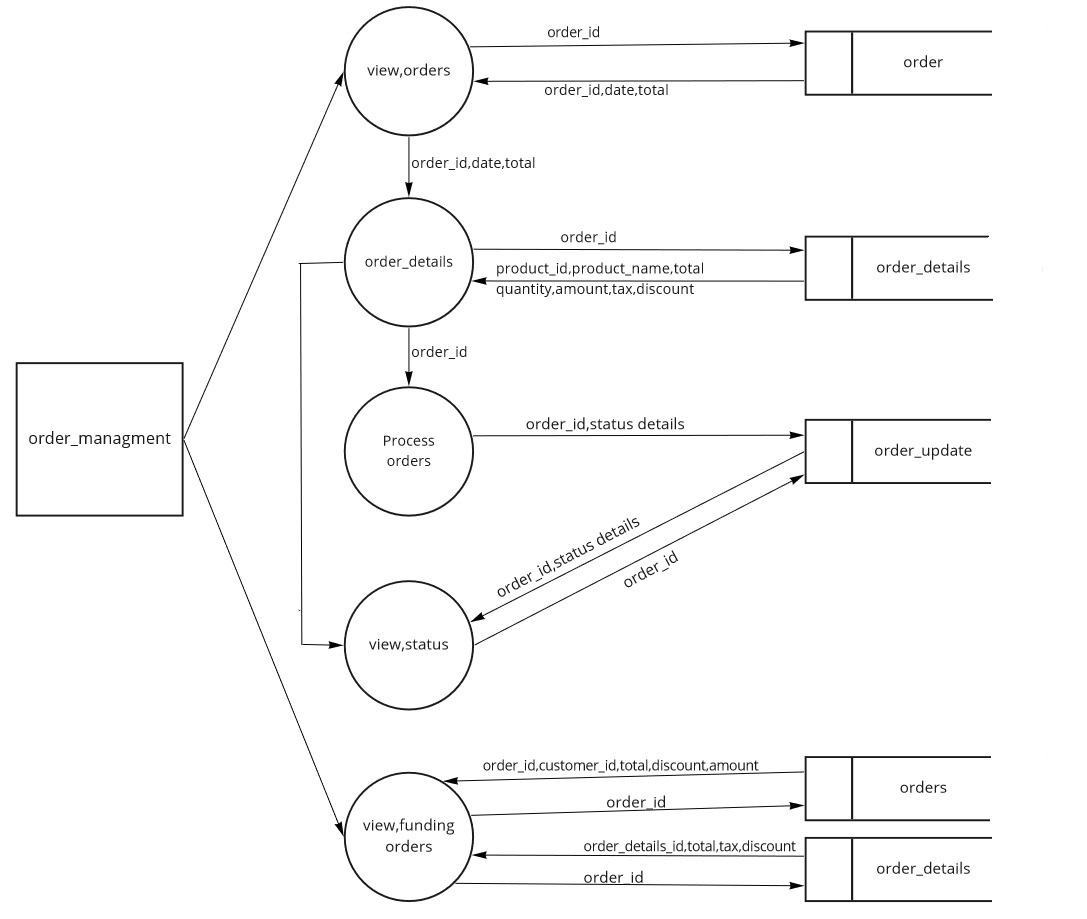


Figure 5.7 Order management (level 1.0)

Level 1.0 Admin(Inventory)

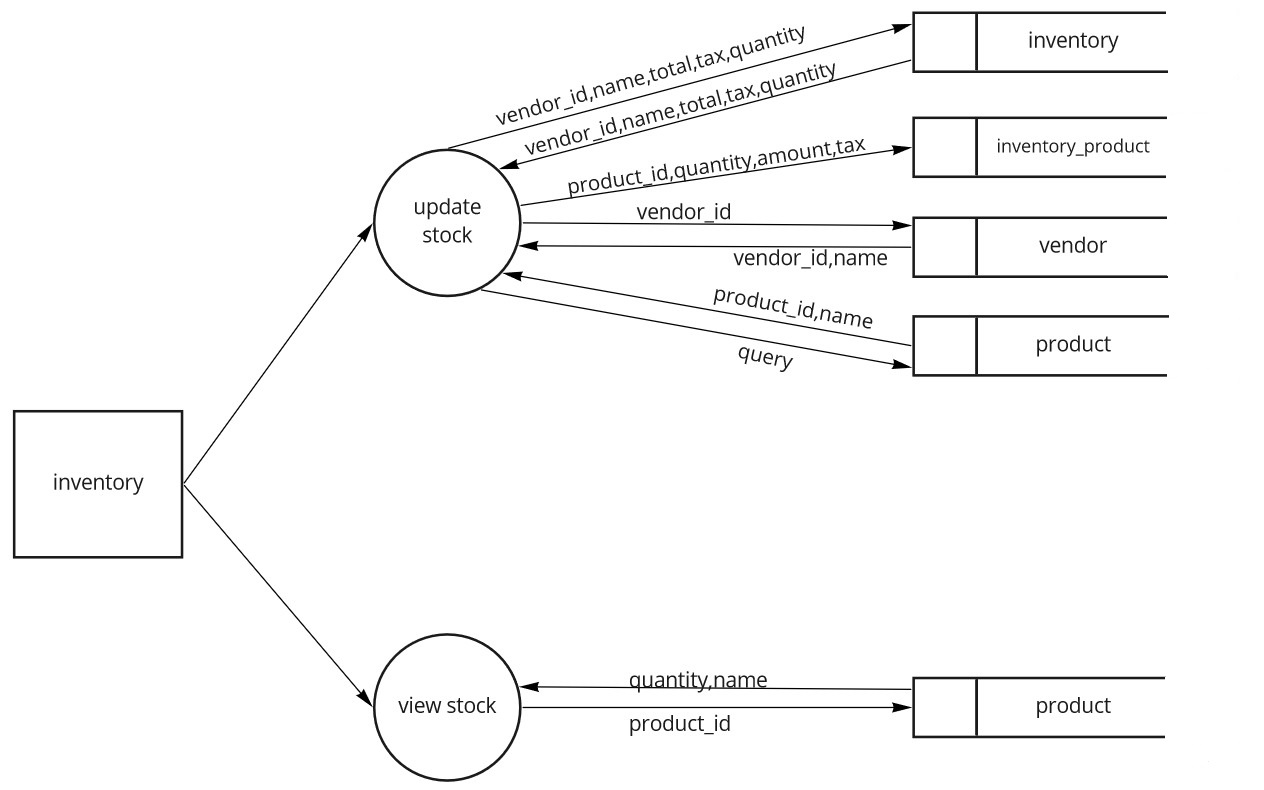


Figure 5.8 Inventory (level 1.0)

Level 1.0 Admin (vendor\_management)

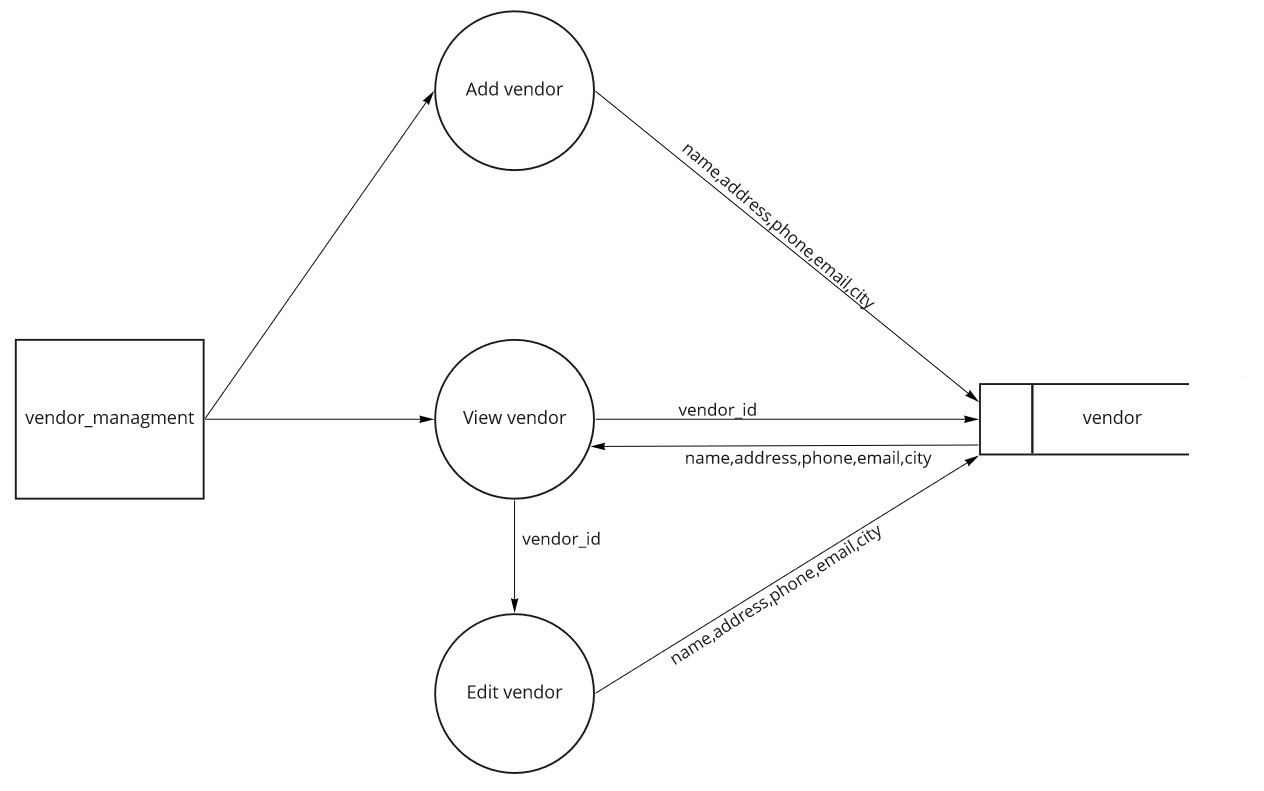
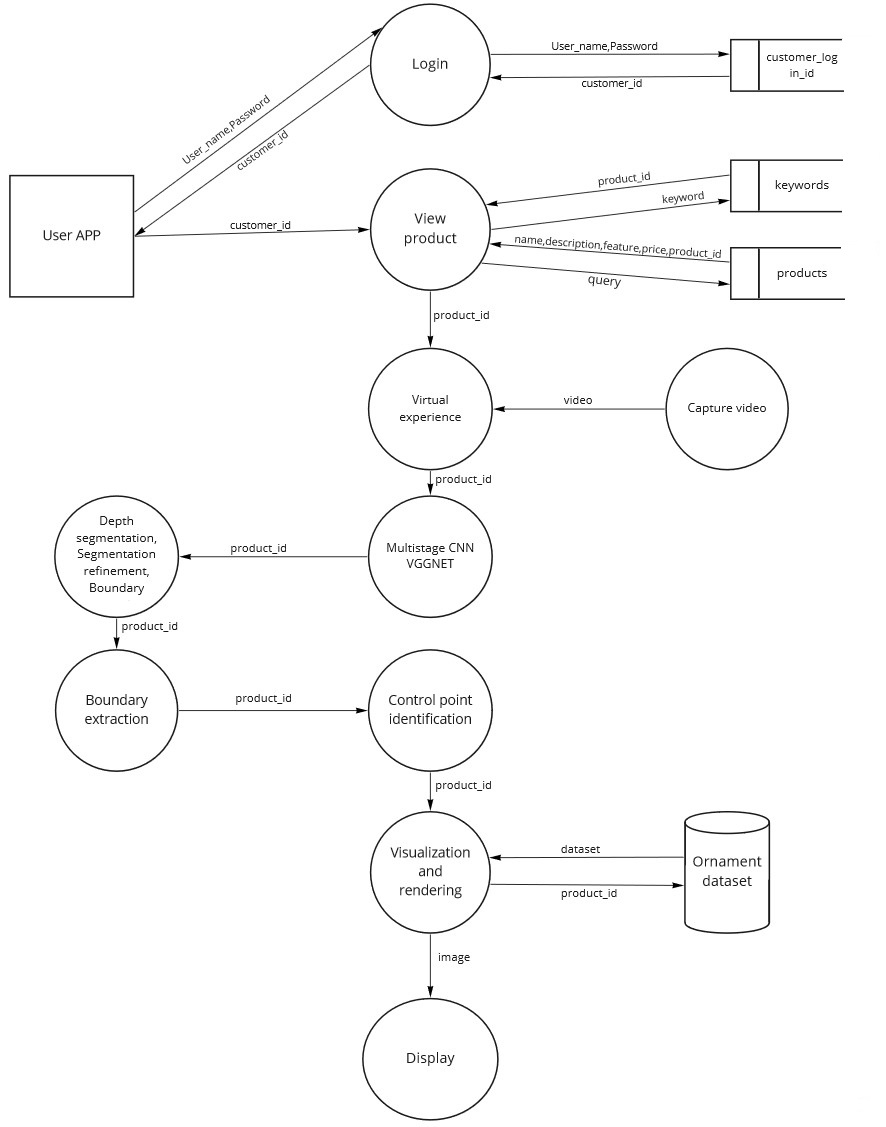


Figure 5.9 Vendor management (level 1.0)

Level 1.1 User

 Figure 5.10 User (level 1.1)

5.3 Database Design

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. The main objectives of databases are data integrity and data independence.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users

Database Management System (DBMS) allow the data to be protected and organized separately from other resources like hardware, software, and programs. DBMS is a software package, which contains components that are not found other data management packages. The significant of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. That is the difference between the logical and physical data.

The main objectives covered in database design are:

• Controlled redundancy

• Data independence

• Accuracy and integrity

• Privacy and security

• Performance

**Admin**

Table number: 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Admin Primary key: admin\_id | | | |
| Description: Admin id and it contains 3 fields | | | |
| Field name | Data type | Constraints | Description |
| admin\_id | int | Primary key | Admin id |
| username | varchar(20) | not null | Name of the user |
| password | varchar(20) | not null | Password of the admin |

**Customer\_registration**

Table number: 5.2

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Customer registration Primary key: customer\_id | | | |
| Description: Customer registration details and it contains fields 7 fields | | | |
| Field name | Data type | Constraints | Description |
| customer\_id | int | Primary key | Customer id |
| Name | varchar(100) | not null | Name of the customer |
| address | varchar(500) | not null | Address of the customer |
| phone | varchar(15) | not null | Phone number of the customer |
| email | varchar(20) | not null | Email of the customer |
| city | varchar(40) | not null | City name |

**Customer\_login\_id**

Table number: 5.3

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Customer\_login\_id Primary key: user\_login\_id | | | |
| Description: Customer id and it contains 4 fields | | | |
| Field name | Data type | Constraints | Description |
| User\_login\_id | varchar(25) | Primary key | User login id |
| User\_name | Varchar(25) | notnull | User name |
| Password | Varchar(25) | notnull | Password |
| customer\_id | Varchar(25) | Foreign key | Customer id |

**Product**

Table number: 5.4

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Product Primary key:product\_id | | | |
| Description: Product and it contains 8 fields | | | |
| Field name | Data type | Constraints | Description |
| product\_id | int | Primary key | Product id |
| product\_name | varchar(20) | not null | Product name |
| description | text | not null | Description of the product |
| price | double | not null | Price of the product |
| quantity | int | not null | No. of quantity |
| tax | double | not null | Tax of the product |
| category\_id | int | not null | Category id |

**Product\_details**

Table number: 5.5

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Product details Primary key: product\_details\_id | | | |
| Description: Product details and it contains 6 fields | | | |
| Field name | Data type | Constraints | Description |
| product\_details\_id | int | Primary key | Product details |
| product\_id | int | foreign key | Product id |
| weight | double | not null | Weight of the product |
| dimension | varchar(20) | not null | Dimensions of the product |
| rating | double | not null | Rating of the product |
| certification | varchar(10) | not null | Certificates |

**Product\_keywords**

Table number: 5.6

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Product keywords Primary key: keyword\_id | | | |
| Description: Product keywords and it contains 4 fields | | | |
| Field name | Data type | Constraints | Description |
| keyword\_id | int | Primary key | Keyword ID |
| Product\_id | int | Foreign key | Product ID |
| keyword | varchar(50) | not null | Keyword |
| description | text | not null | Description |

**Category**

Table number: 5.7

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Category Primary key: category\_id | | | |
| Description: Product keywords and it contains 3 fields | | | |
| Field name | Data type | Constraints | Description |
| category\_id | int | Primary key | Category ID |
| category | varchar(30) | not null | Category name |
| feature | varchar(30) | not null | Features |

**Order**

Table number: 5.8

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Order Primary key: order\_id | | | |
| Description: Order and it contains 8 fields | | | |
| Field name | Data type | Constraints | Description |
| order\_id | int | Primary key | Order ID |
| customer\_id | int | Foreign key | Customer ID |
| date | date | not null | Date |
| total | double | not null | Total number |
| discount | double | not null | Discount amount |
| amount | double | not null | Amount |
| status | Varchar(20) | not null | Status of the product |

**Order\_details**

Table number: 5.9

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Order details Primary key: order\_details\_id | | | |
| Description: Order details and it contains 7 fields | | | |
| Field name | Data type | Constraints | Description |
| order\_details\_id | int | Primary key | Order details ID |
| order\_id | int | Foreign key | Order ID |
| product\_id | int | Foreign key | Product ID |
| Total | double | not null | Total |
| tax | double | not null | Tax amount |
| discount | double | not null | Discount amount |

**Order\_update**

Table number: 5.10

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Order update Primary key: order\_update\_id | | | |
| Description: Order update id and it contains 5 fields | | | |
| Field name | Data type | Constraints | Description |
| order\_update\_id | int | Primary key | Order update |
| order\_id | int | Foreign key | Order ID |
| status | varchar(20) | not null | Status |
| details | text | not null | Details of the order |
| date | date | not null | Ordered date |

**Customer\_feedback**

Table number: 5.11

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Customer feedback Primary key: feedback\_id | | | |
| Description: Customer feedback and it contains 4 fields | | | |
| Field name | Data type | Constraints | Description |
| feedback\_id | int | Primary key | Feedback ID |
| customer\_id | int | Foreign | Customer ID |
| Feedback | text | not null | Customer feedback |

**Vendor**

Table number: 5.12

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: vendor Primary key: vendor\_id | | | |
| Description: Vendor details and it contains 7 fields | | | |
| Field name | Data type | Constraints | Description |
| vendor\_id | int | Primary key | Vendor ID |
| name | varchar(30) | not null | Name of the vendor |
| address | varchar(100) | not null | Address |
| phone | varchar(13) | not null | Phone number |
| email | varchar(30) | not null | Email ID |
| city | varchar(50) | not null | Name of the city |

**Inventory**

Table number: 5.13

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Inventory Primary key: inventory\_id | | | |
| Description: Inventory and it contains 5 fields | | | |
| Field name | Data type | Constraints | Description |
| inventory\_id | int | Primary key | Inventory ID |
| vendor\_id | int | Foreign key | Vendor\_ID |
| quantity | int | not null | Number of quantity |
| amount | double | not null | Amount |
| tax | double | not null | Tax |

**Inventory\_products**

Table number: 5.14

|  |  |  |  |
| --- | --- | --- | --- |
| Table name: Inventory products Primary key: in\_product\_id | | | |
| Description: Inventory products and it contains 5 fields | | | |
| Field name | Data type | Constraints | Description |
| in\_product\_id | int | Primary key | Inventory Product ID |
| product\_id | int | Foreign key | Product ID |
| quantity | Int | not null | Number of quantity |
| amount | double | not null | Amount |

5.4 Input design

Input design is the method by which valid data are accepted from the user. This part of the designing requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify these errors. The input design is the process of converting the user-oriented description of inputs into a programmer-oriented specification. The objective of input design is to create an input layout that is easy to follow and prevents the user from committing errors. It covers all phases of input, right from the creation of initial databases to the actual data entry into the system. The input design is the link that ties the system into the world of its users. Hence, lays its importance in the design phase. The input design makes sure that while entering data, the end-users understand the format in which the data is to be entered so that it is accepted by the system, the data values that are mandatory for the system to function, the order in which transactions need to be processed etc. The goal designing input data is to make the automation as easy and free from errors as possible. Input design consists of the following processes:-

* Designing graphical user entry screen is easy to use. Designing procedures and functions to valid the data as per business rules.
* Designing functions needed to store data into a usable form for processing.
* Designing the common integrated functions that can be used by all other users when needed.

5.5 Output design

A quality output is one, which meets the requirement of the end user and presents the information clearly. In any system results of processing are communicated to the user and to other system through output. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output designs improve the system’s relationship to help user decision-making.

Output forms are the forms where the respective module. We may also see the outputs in reports. The reports often refresh for every change in the database. We can create reports by using SQL queries. Outputs form the system are requires to communicate the results of processing to users. Formats of outputs are defined during output design. The success of the system depends on how well the output reports are generated.

Designing output the following things are to be considered.

* Determine what information to present.
* Arrange the presentation of information in an acceptable format.
* Decide how to distribute the output to intend receipts.
* Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy.
* Efficient and intelligent output design should improve system relation with the user and help in decision making that is, this makes system user friendly to be displayed or printed as per the user’s choice.

Table number 5.15

|  |  |  |
| --- | --- | --- |
| **Process** | **Input design** | **Output design** |
| Login page for admin | Enter user name, password | Show home page |
| Login page for user | Enter user name, password | Show home page |
| Add product | Enter product details | Product added successfully |
| Add category | Enter category details | Category added successfully |
| Edit product | Select the product and edit | Edit completed |
| View product | Select category and click product | Show product and it’s details |
| View order | Select ordered product | Show order status |
| Update stock | Enter stock details | Stock updated |
| View stock | Enter product details | Show stock quantity |
| Add vendor | Enter vendor details | Vendor added successfully |
| View vendor | Enter vendor id | Show vendor |
| Edit vendor | Enter vendor id | Edit completed |

5.6 Program design

1. Admin

Step 1: Start

Step 2: Once logged in the Admin has the privilege to manage products, orders, inventory, and vendor.

Step 3: In product function admin can add new product, add category, update pictures of the product, view product and also can edit product.

Step 4: In order function admin can view orders, order details, process orders, view status and also can view funding of the orders.

Step 5: In inventory function admin can update stock and view stock details.

Step 6: In vendor function admin can add vendor, view vendor and edit vendor.

Step 7: Stop

1. User

Step 1: Start

Step 2: Once logged in the customer can view categories, view products in augmented reality, book products, billing.

Step 3: Customer can buy products, can cart products, view or edit cart and also can give phone number, email and address for updates and product delivery.

Step 4: Can add credit/debit card details. Billing can also be done.

Step 5: Can view order status.

Step 6: Stop

CHAPTER 6

FUNCTIONAL AND NON FUNCTIONAL

REQUIREMENTS

6.1 Functional Requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally, functional requirements are expressed in the form "system must do requirement ".

Functional requirements for each of the uses cases described below:

* User should register and login for shopping to make a virtual trail of the ornaments on them.
* Ornaments images with specified dimension should be uploaded to get virtual experience.
* Admin should add stock update to view inventory.

6.2 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-functional requirements are “system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”, “quality goals", "quality of service requirements" and "non-behavioral requirements.

Some of the non-functional requirements are mentioned below:

i. **Usability**: The system shall have a clean interface with only needed features, clear terminology and tool tips wherever necessary. Warnings or alerts shall be specified in clear way.

ii. **Efficiency**: The system shall respond to different searches being conducted like searching particular product, search quantity, etc. in a very fast way.

iii. **Interoperability**: The system shall be able to interact with other systems. The system should able to be supported at least one software which has a relationship with payment process

iv. **Portability**: The system shall be independent of the specific technological platform used to implement it.

v. **Reliability**: Reliability defined as a measure of the time between failures occurring in a system (measure show frequently the system fails), so that the system shall operate without any failure for a particular period of time

vi. **Availability**: Availability measures the percentage of time the system is in its operational state so that the system shall be available for use 24 hours per day and 365days per year.

CHAPTER 7

TESTING

7.1 Testing strategies

An engineered product can be tested in one of these two ways. These testing strategies include:

* Black box testing
* White box testing

**White box testing**

White-box testing is a method of testing the application at the level of the source code. White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are chooses inputs to exercise paths through the code and determine the expected outputs.

**Black box testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

7.2 Unit testing

In computer programming, unit testing is a software method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures ,are tested to determine whether they are fit for use intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.

Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. If forms the basis for component testing.

In the project each module is tested individually and is found to be an error free one.

7.3 Integration testing

This is the final step in testing. In this case all the modules were combined and given the test data. The combined module works successfully without any side effect on other programs. Everything was found to be working correctly.

In this the entire system was tested as a whole with all modules. This form of testing is popularly known as Black Box testing or system testing. Black Box testing methods focus on the functional requirement of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. 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, performance errors and initialization errors and termination errors.

In this project each module is tested individually and all the modules are integrated together and the integration testing is carried out for the whole system. The whole system is working accurately without any errors.

7.4 System testing

Software testing is critical element of software quality assurance and represents the ultimate review of specifications, design and code generation. System testing is the stage of implementation, it is aimed for ensuring that the system works accurately and efficiently before live operations commences. Nothing is complete without testing, as it is vital success of the system.

Testing Objectives:

There are several rules that can serve as testing objectives, they are

• Testing is a process of executing a program with the intent of finding an error

• A good test case is one that has high probability of finding an undiscovered error.

• A successful test is one that uncovers an undiscovered errors.

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for used to design test cases. The tester effective and consistent regression testing.

7.5 Testing Results

* Admin login

Table number: 7.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test scenario | Test steps | Test data | Expected  Results | Actual  Results | Pass/fail |
| 1 | Check admin  Login with valid data | Go to site Enter admin id, Enter password  Click sign in | Admin id=admin  Password=  Password12 | Admin should login into application | As expected | Pass |
| 2 | Check admin login with invalid data | Go to site Enter admin id, Enter password  Click sign in | Admin id=user  Password=  12345 | Admin login fail into application | As expected | Pass |
| 3 | Check admin login with valid admin id and invalid password | Go to site Enter admin id, Enter password  Click sign in | Admin id=admin  Password=  12345 | Admin login fail into application | As expected | Pass |
| 4 | Check admin login with invalid admin id and valid password | Go to site Enter admin id, Enter password  Click sign in | Admin id=user  Password=  Password12 | Admin login fail into application | As expected | Pass |

* Customer login id

Table number: 7.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test scenario | Test steps | Test data | Expected  Results | Actual  Results | Pass/fail |
| 1 | Check User  Login with valid data | Go to site Enter user id, Enter password  Click sign in | User id=user  Password=  Password12 | User should login into application | As expected | Pass |
| 2 | Check user login with invalid data | Go to site Enter user id, Enter password  Click sign in | User id=admin  Password=  12345 | User login fail into application | As expected | Pass |
| 3 | Check user login with valid user id and invalid password | Go to site Enter user id, Enter password  Click sign in | User id=user  Password=  12345 | User login fail into application | As expected | Pass |
| 4 | Check user login with invalid user id and valid password | Go to site Enter user id, Enter password  Click sign in | User id=admin  Password=  Password12 | User login fail into application | As expected | Pass |

* Add Product

Table number: 7.3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test scenario | Test steps | Test data | Expected  Results | Actual  Results | Pass/fail |
| 1 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=Alor Diamond bangle, Features=Diamond  Description=Antique model, Price=50000,Tax in %=10,Quantity=25 | Product added successfully | As expected | Pass |
| 2 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=Alor Diamond bangle, Features=,  Description=Antique model, Price=50000,Tax in %=10,Quantity=25 | Product added failed | As expected | Pass |
| 3 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=Alor Diamond bangle, Features=Diamond  Description=, Price=50000,Tax in %=10,Quantity=25 | Product added failed | As expected | Pass |
| 4 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=Alor Diamond bangle, Features=Diamond  Description=Antique model, Price=,Tax in %=10,Quantity=25 | Product added failed | As expected | Pass |
| 5 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=Alor Diamond bangle, Features=Diamond  Description=Antique model, Price=50000,Tax in %=,Quantity=25 | Product added failed | As expected | Pass |
| 6 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=Alor Diamond bangle, Features=Diamond  Description=Antique model, Price=50000,Tax in %=10,Quantity= | Product added failed | As expected | Pass |
| 7 | Check product with valid details for product name, features, description, price, tax in %, quantity | Go to site Enter product name, features, description, price, tax in %, quantity | Product name=, Features=Diamond  Description=Antique model, Price=50000,Tax in %=10,Quantity=25 | Product added failed | As expected | Pass |

* Add Vendor

Table number: 7.4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test scenario | Test steps | Test data | Expected  Results | Actual  Results | Pass/fail |
| 1 | Check add vendor with valid details for Name, Address, Phone, email, City | Go to site Enter Name, Address, Phone, email, City | Name=Alex wick, Address=Street venue, Tulsa, Phone=9748224599, email=alx@outlook.com, City=Tulsa | Vendor added successfully | As expected | Pass |
| 2 | Check add vendor with valid details for Name, Address, Phone, email, City | Go to site Enter Name, Address, Phone, email, City | Name=Alex wick, Address=, Phone=9748224599, email=alx@outlook.com, City=Tulsa | Vendor adding failed | As expected | Pass |
| 3 | Check add vendor with valid details for Name, Address, Phone, email, City | Go to site Enter Name, Address, Phone, email, City | Name=Alex wick, Address=Street venue, Tulsa, Phone=, email=alx@outlook.com, City=Tulsa | Vendor adding failed | As expected | Pass |
| 4 | Check add vendor with valid details for Name, Address, Phone, email, City | Go to site Enter Name, Address, Phone, email, City | Name=Alex wick, Address=Street venue, Tulsa, Phone=9748224599, email=, City=Tulsa | Vendor adding failed | As expected | Pass |
| 5 | Check add vendor with valid details for Name, Address, Phone, email, City | Go to site Enter Name, Address, Phone, email, City | Name=Alex wick, Address=Street venue, Tulsa, Phone=9748224599, email=alx@outlook.com, City= | Vendor adding failed | As expected | Pass |
| 6 | Check add vendor with valid details for Name, Address, Phone, email, City | Go to site Enter Name, Address, Phone, email, City | Name=, Address=Street venue, Tulsa, Phone=9748224599, email=alx@outlook.com, City=Tulsa | Vendor adding failed | As expected | Pass |

* Update stock

Table number: 7.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test scenario | Test steps | Test data | Expected  Results | Actual  Results | Pass/fail |
| 1 | Check update stock with valid details for Vendor id, Product name, Tax in %, Total, Quantity | Go to site Enter Vendor id, Product name, Tax in %, Total, Quantity | Vendor id=Adam traders, Product name=Droplets stud, Tax in %=15, Total=165, Quantity=2 | Stock update successfully | As expected | Pass |
| 2 | Check update stock with valid details for Vendor id, Product name, Tax in %, Total, Quantity | Go to site Enter Vendor id, Product name, Tax in %, Total, Quantity | Vendor id=Adam traders, Product name=, Tax in %=15, Total=165, Quantity=2 | Stock update fail | As expected | Pass |
| 3 | Check update stock with valid details for Vendor id, Product name, Tax in %, Total, Quantity | Go to site Enter Vendor id, Product name, Tax in %, Total, Quantity | Vendor id=Adam traders, Product name=Droplets stud, Tax in %=, Total=165, Quantity=2 | Stock update fail | As expected | Pass |
| 4 | Check update stock with valid details for Vendor id, Product name, Tax in %, Total, Quantity | Go to site Enter Vendor id, Product name, Tax in %, Total, Quantity | Vendor id=Adam traders, Product name=Droplets stud, Tax in %=15, Total=, Quantity=2 | Stock update fail | As expected | Pass |
| 5 | Check update stock with valid details for Vendor id, Product name, Tax in %, Total, Quantity | Go to site Enter Vendor id, Product name, Tax in %, Total, Quantity | Vendor id=Adam traders, Product name=Droplets stud, Tax in %=15, Total=165, Quantity= | Stock update fail | As expected | Pass |
| 6 | Check update stock with valid details for Vendor id, Product name, Tax in %, Total, Quantity | Go to site Enter Vendor id, Product name, Tax in %, Total, Quantity | Vendor id=, Product name=Droplets stud, Tax in %=15, Total=165, Quantity=2 | Stock update fail | As expected | Pass |

CHAPTER 8

RESULT AND DISCUSSION

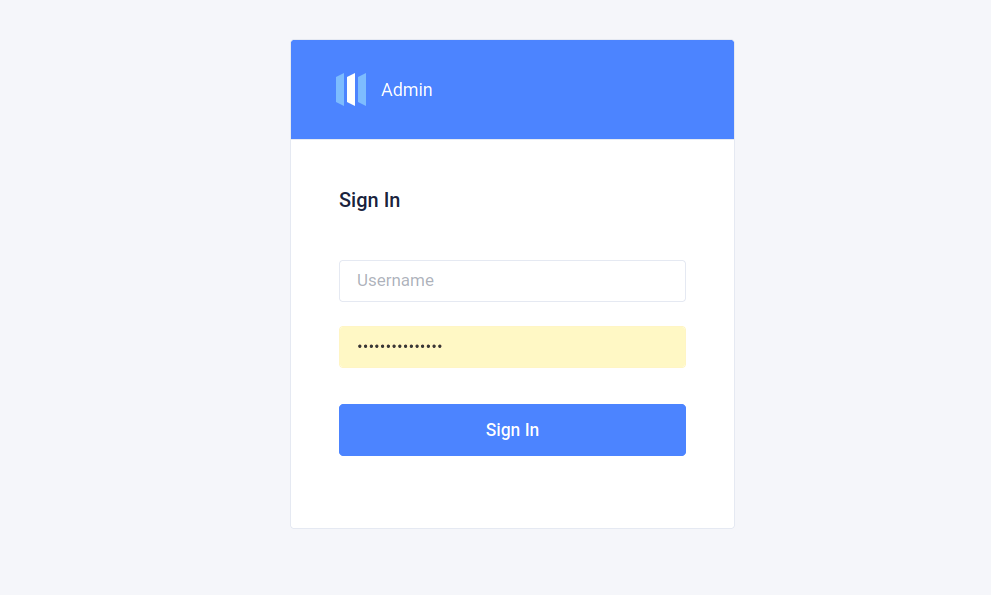
8.1 Results (Salient features)

The main motivation and objective of this system is to provide a solution to reduce inconvenience in finding a better ornament for the customer through augmented reality experience. Enabling the user experiencing a friendly user interface. Systematic handling of the schedules in such a way is a key to increasing its manageability and its competence. The proposed system incorporated with the following features.

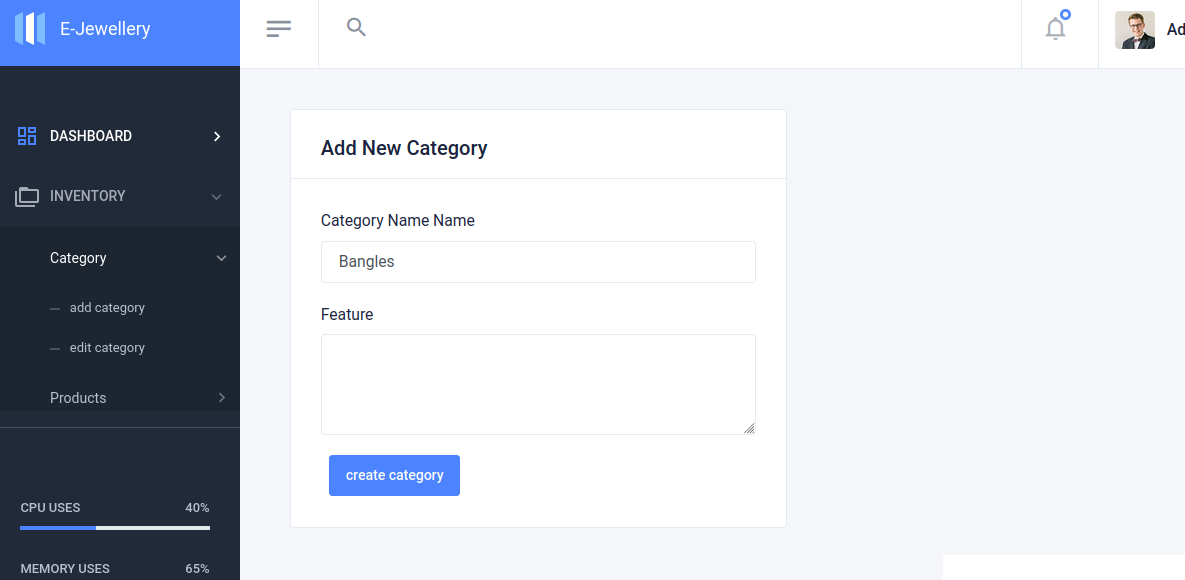
* Human effort can be reduced.
* Improved efficiency.
* Easy access to the machine related documents
* Efficient management of the relationship between the main asset and its sub-components, modules and parts
* Dynamic tracing of the status of maintenance, depending on whether the maintenance action is planned or performed.

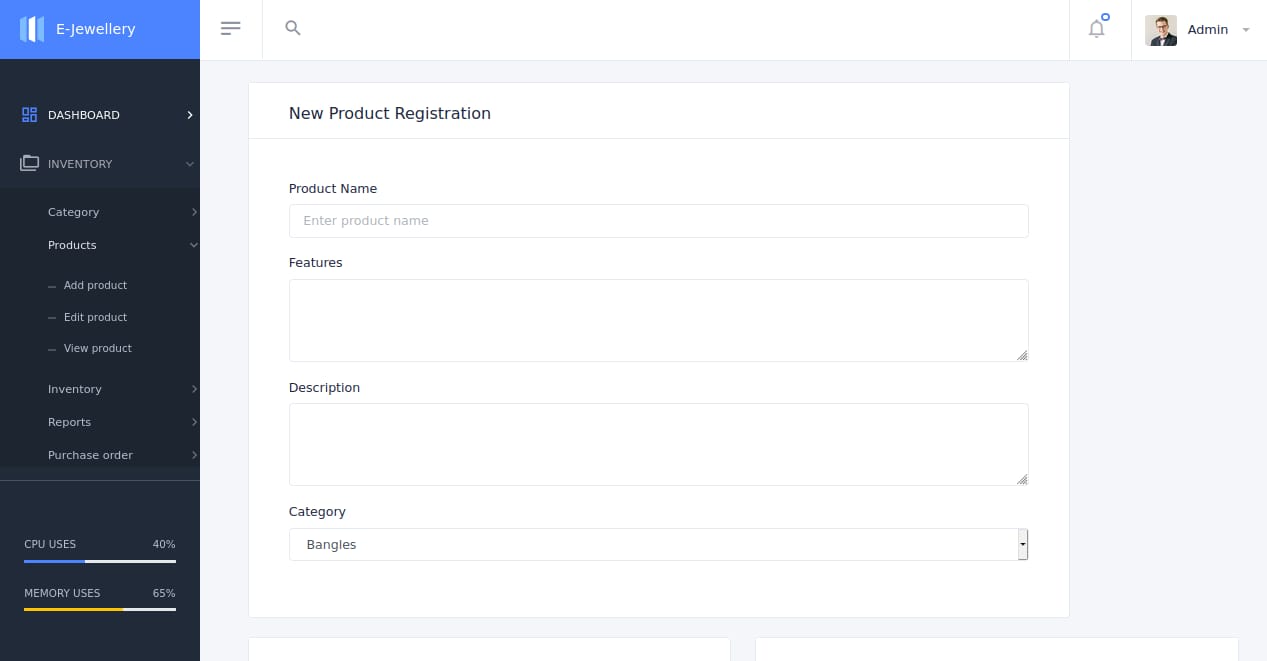
8.2 Screen shots

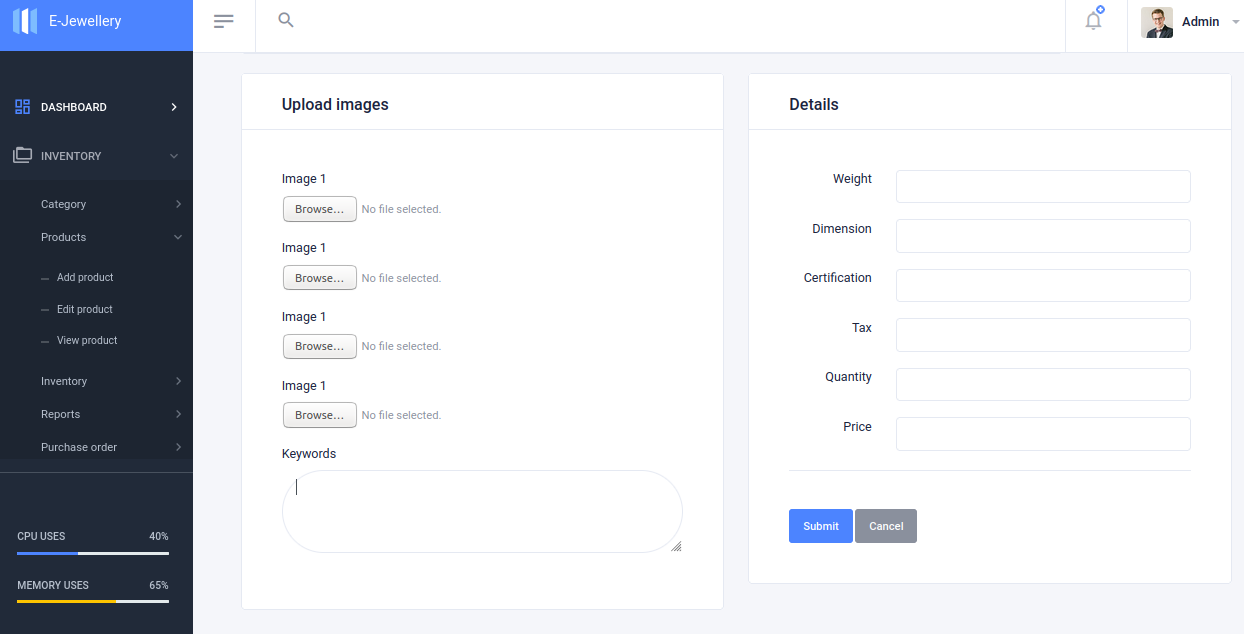
1. Admin login



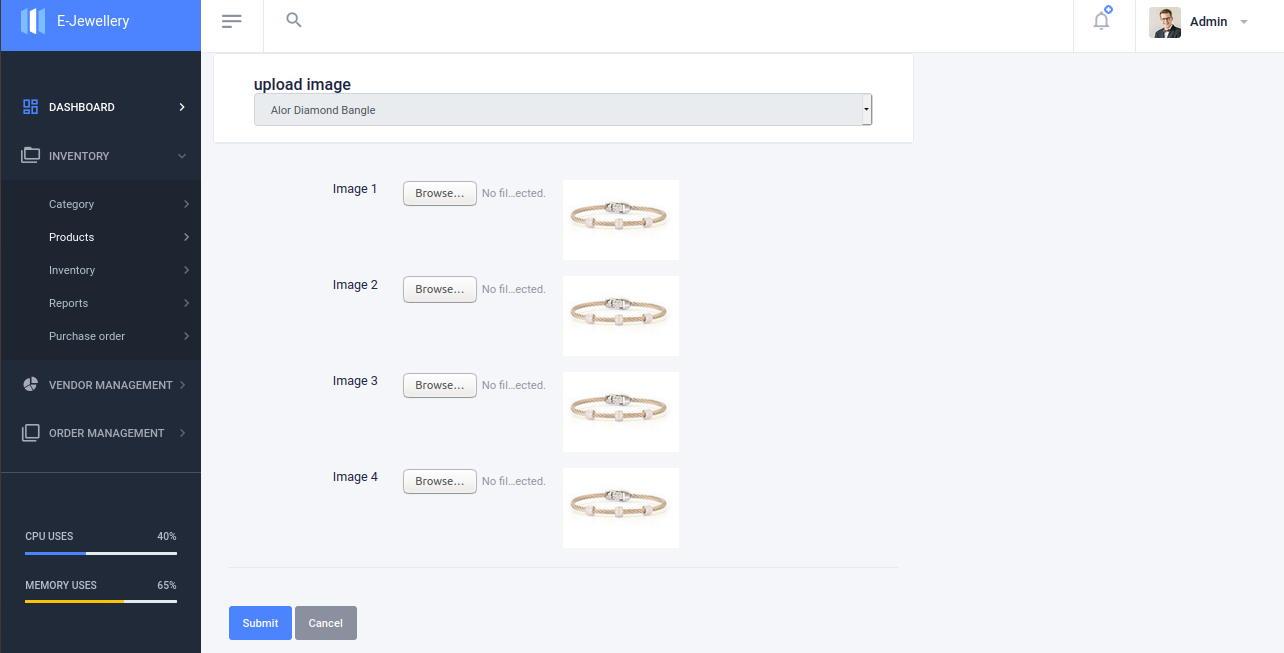
1. Add Category



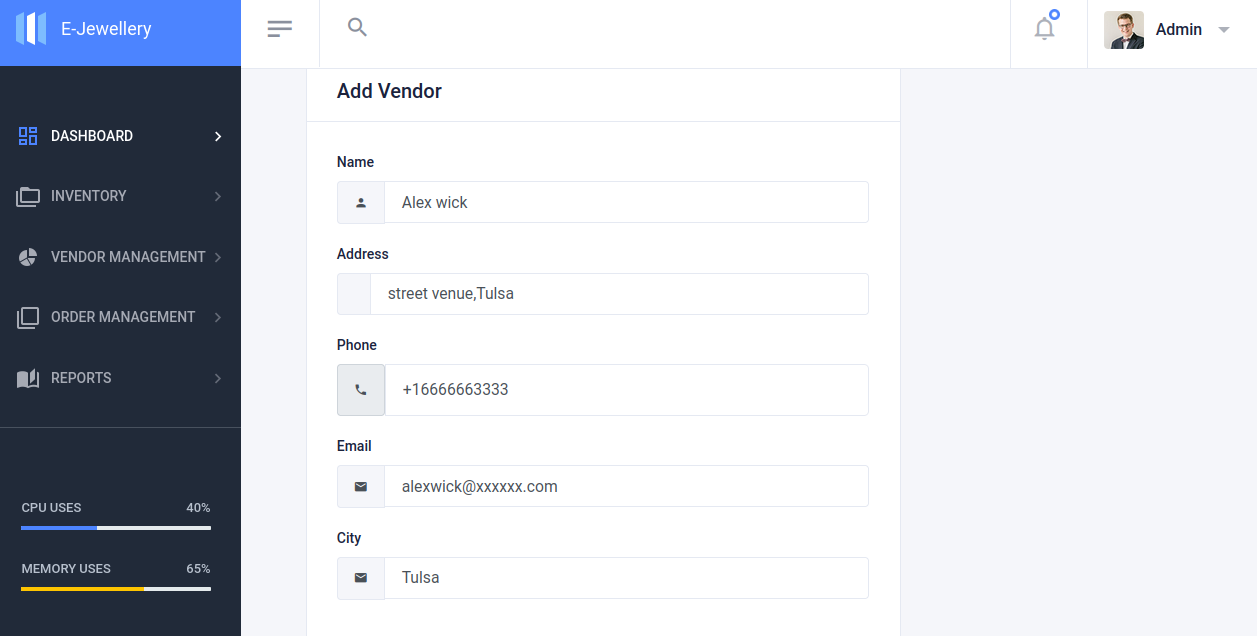
1. Add Product



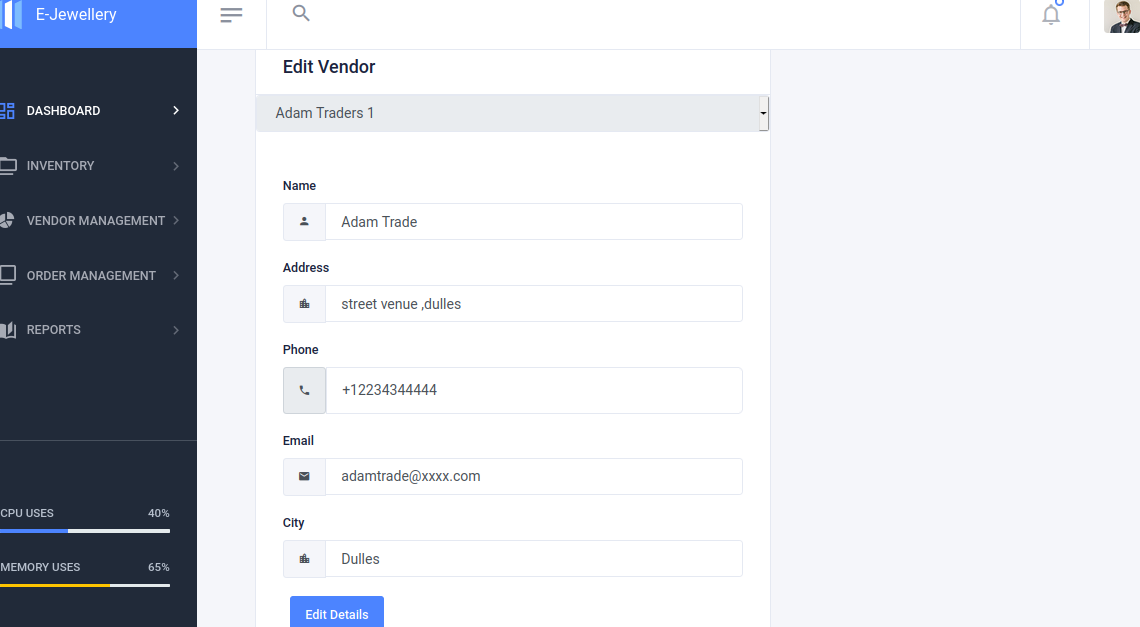
1. Upload picture



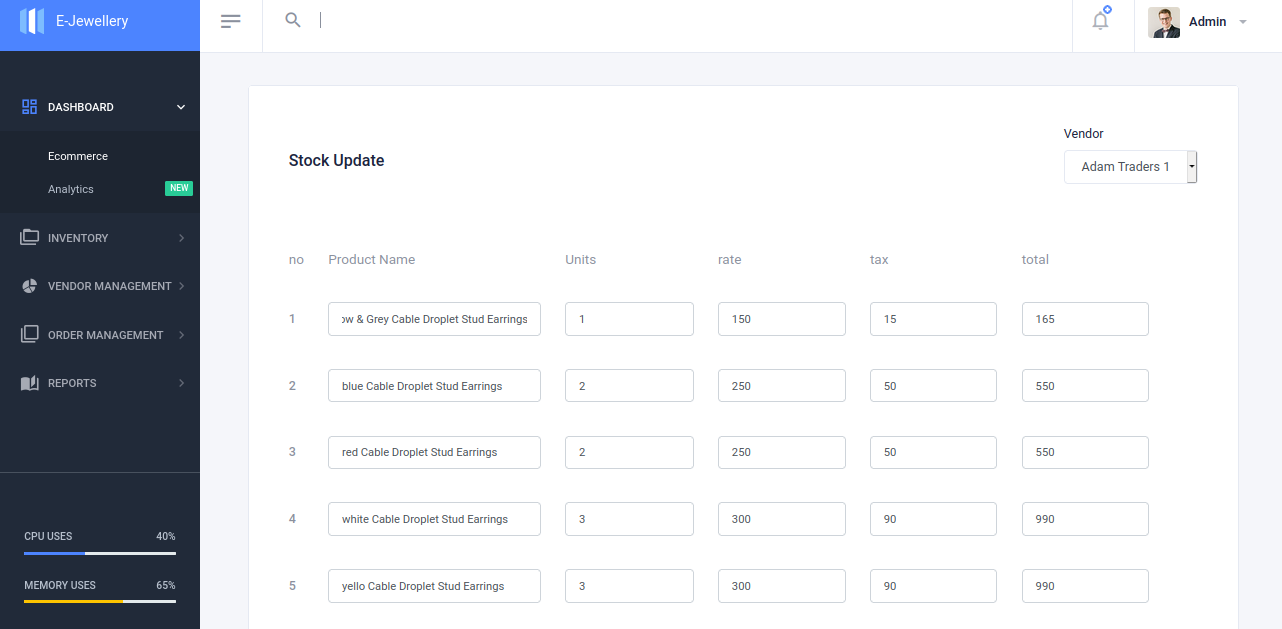
1. Add Vendor



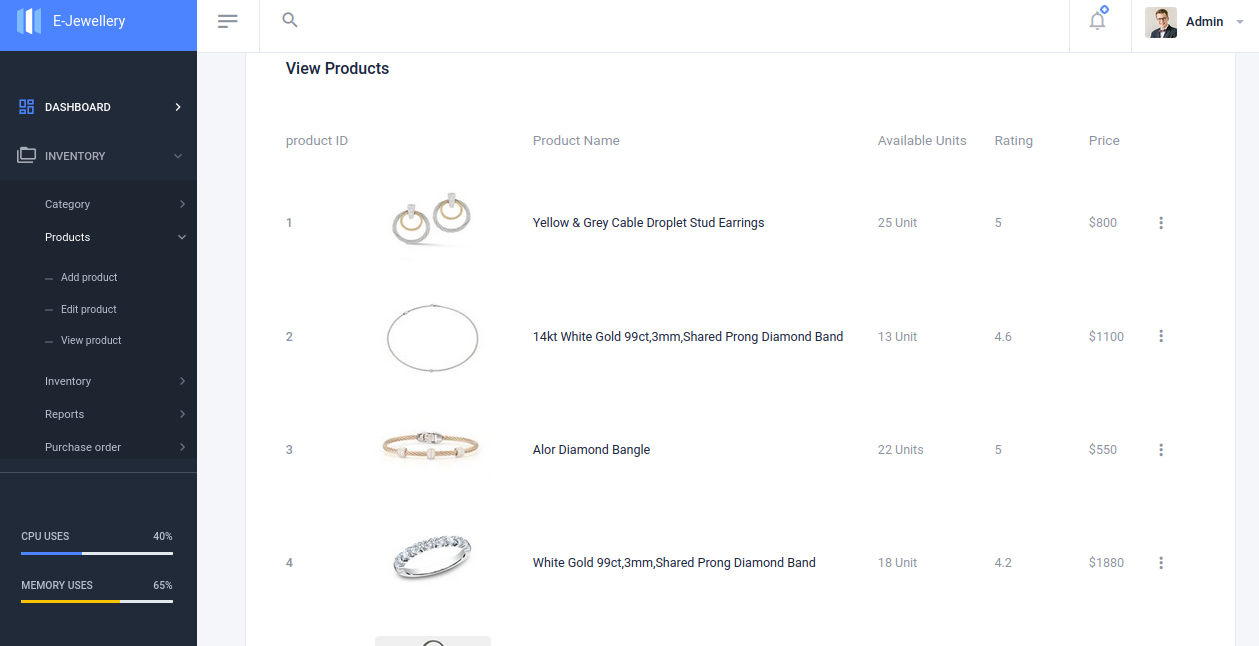
1. Edit Vendor



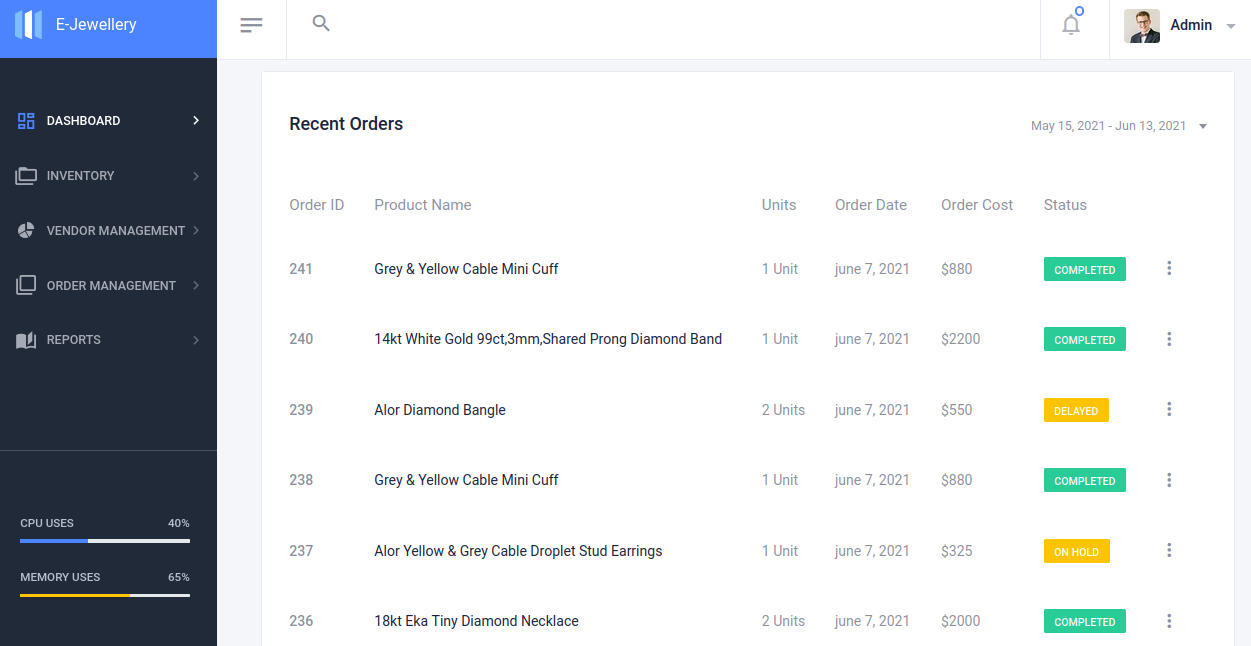
1. Update stock



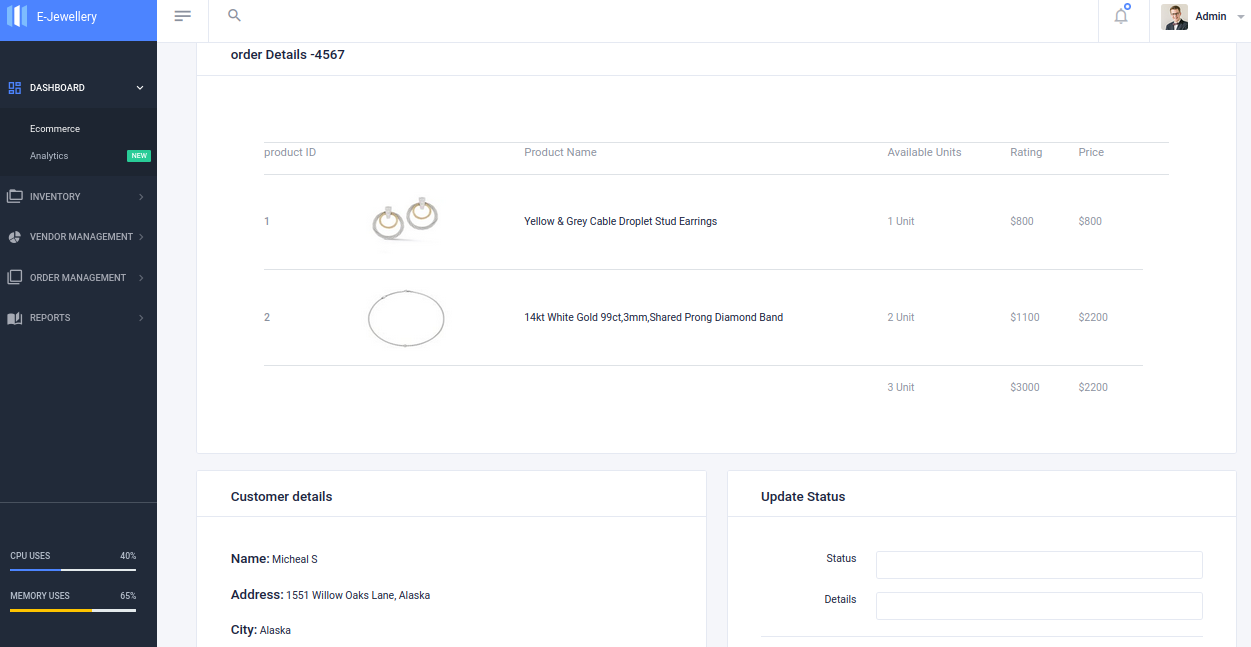
1. View product



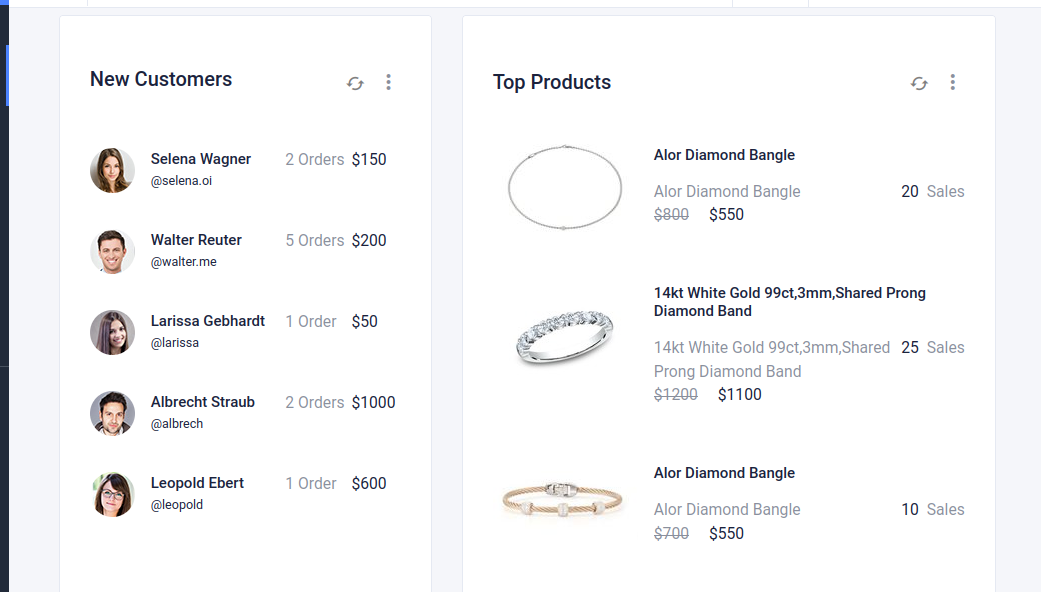
1. View orders



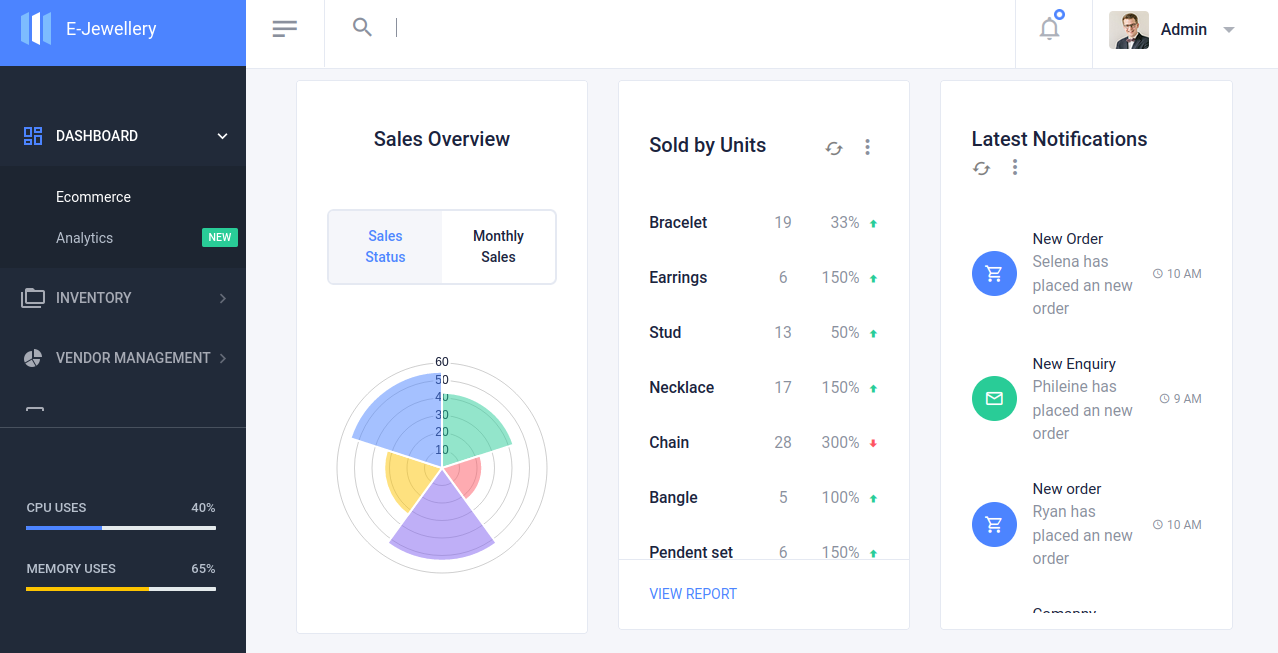
1. Order update



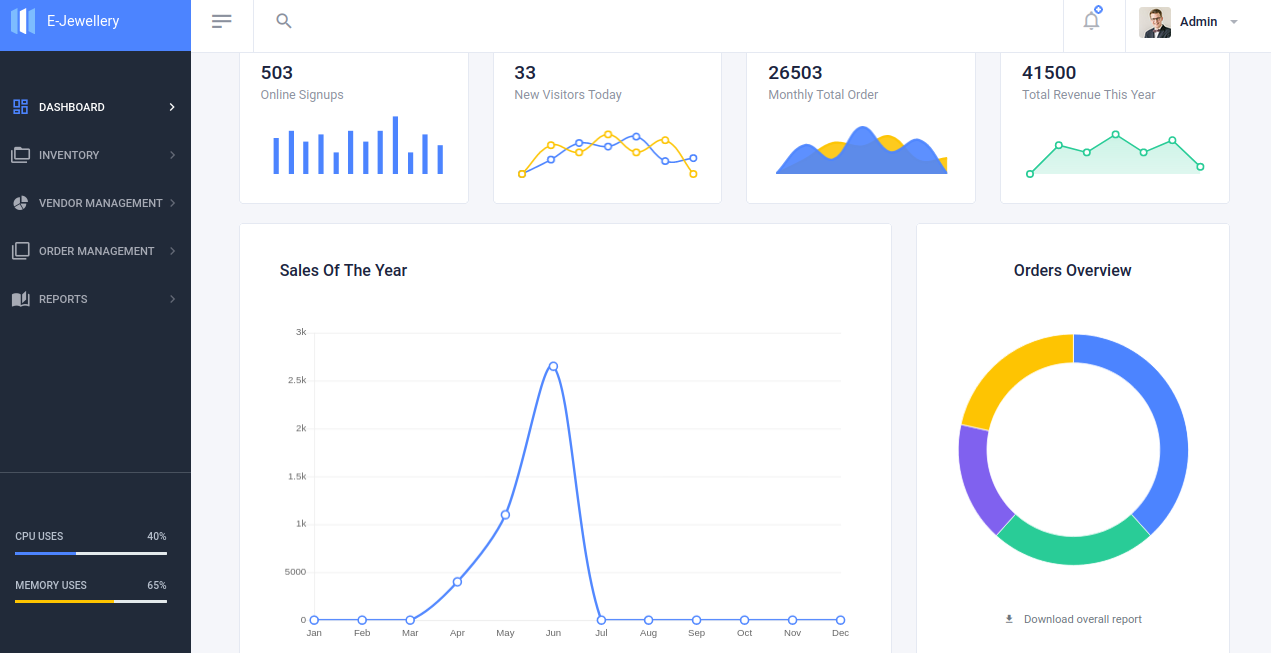
1. Dash board 1



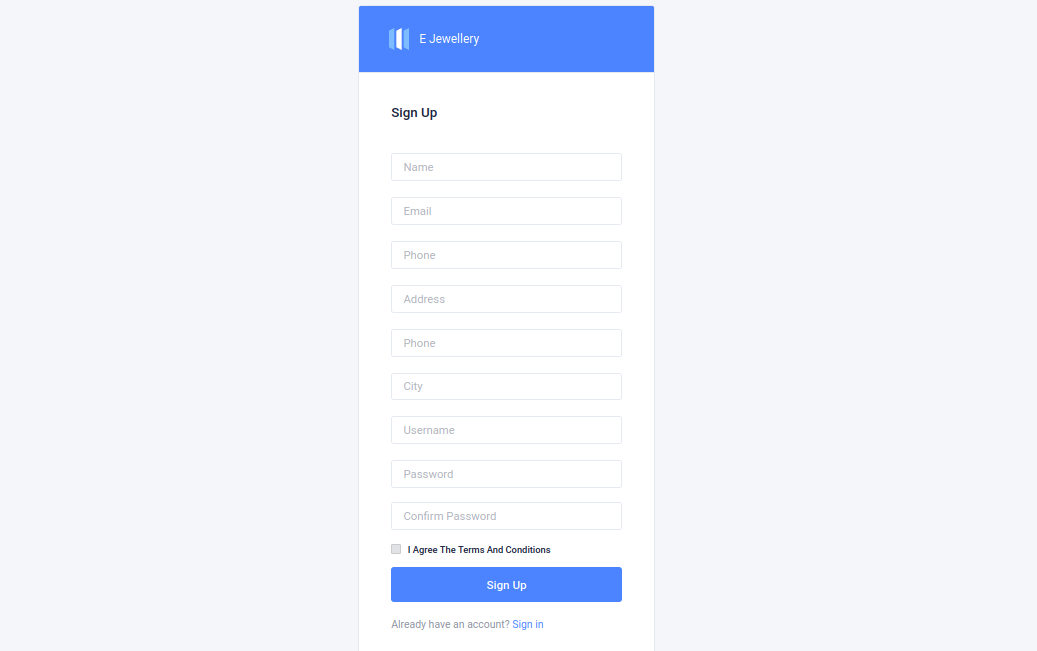
1. Dash board 2



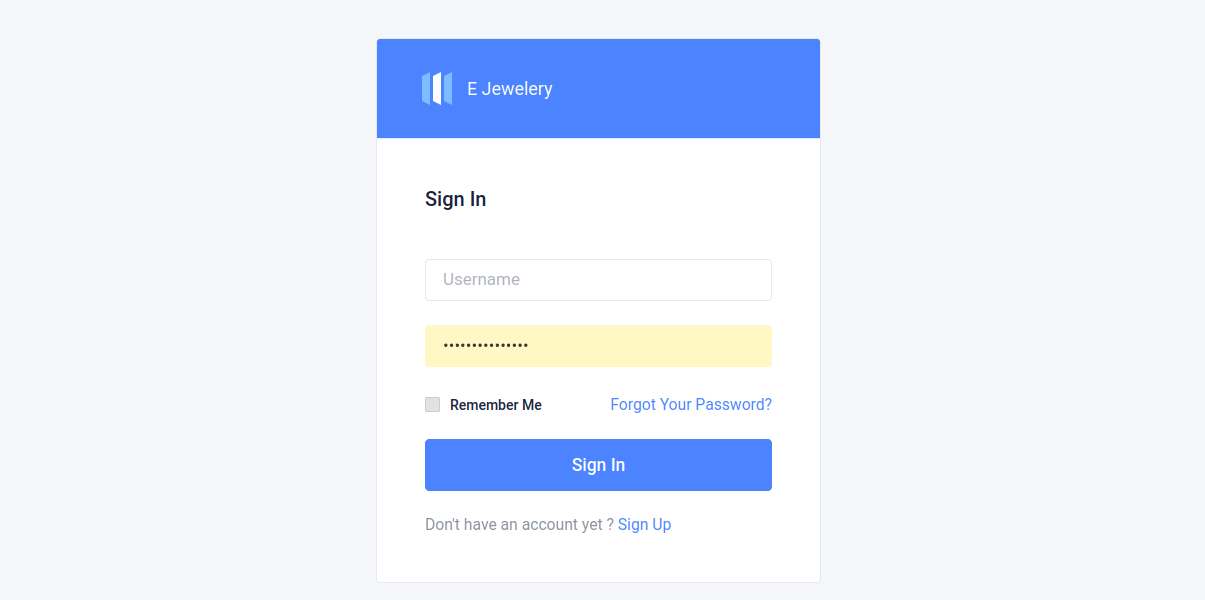
1. Dash board 3



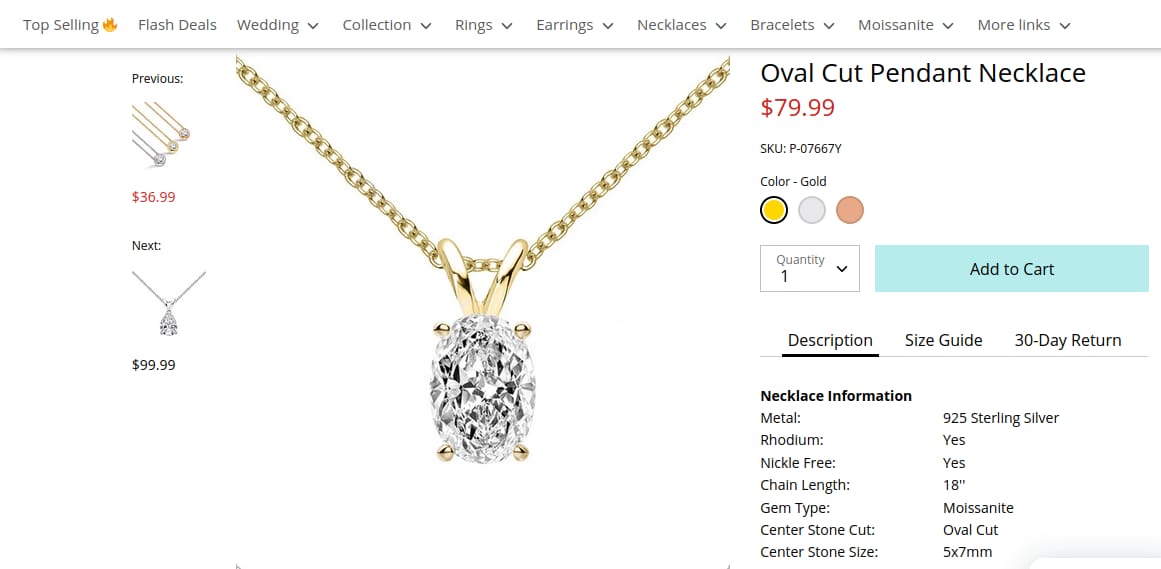
1. User sign-up



1. User login



1. Product view



1. Preview of product in AR



CHAPTER 9

CONCLUSION

9.1 System Implementation

The purpose of System implementation as making the new system making available to be prepared set of users and positioning ongoing support and maintenance of the system within the performing organization. At a final level of detail, deploying the system consists of executing all steps necessary to educate the consumers on the use of a new system, placing the newly developed system into production, continuing that all data required at the start of the operations is available and accurate, and validating that business functions that interact with the system are functioning properly.

The implementation involves following things:

• Careful planning.

• Investigation of the system considerations.

• Design the method to achieve the changeover.

• Evaluation of change over method.

There are three types of implementations:

• Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.

• Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned, there can be many problems. Some large computer systems have taken as long as a year to convert.

• Implementation of a modified application to replace the existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files. Every system requires periodic evaluation after implementation.

This is to review the performance of the system and to evaluate against established standard or criteria. A study is conducted for measuring the performance of the system against pre-defined requirements. This study results a post-implementation review that determines how well the system continues to meet the performance specification.

9.2 Conclusion

In many previous works the human has to adjust himself such that he fits into the given image displayed in the screen. This makes uncomfortable for the user. Thus proposed system is more reliable than the previous existing systems. This application offers customized, pleasant and fun-filled shopping experience for jewellery shoppers. Here, the human face is detected automatically so that the user may not fit in the given image.

This system could be used for enhancing security as it is developed in Augmented Reality. In the application user can view the ornaments in 3D which overcomes occlusion problem also. Hence, using the application user can wear the ornaments virtually, thus proposed system could be more reliable. This makes jewellery safe and moreover avoids the risk of theft.

9.3 Future Enhancement

* This project can be modified at any time by adding any more features and can develop the databases with more new features.
* Future work should focus on increasing the options for watches and sunglasses and incorporating the ability for the users to share their designed rings on social plate form like Facebook, WhatsApp etc. There is also a need for further user testing, which will help to improve the quality of the AR App.
* Mobile application for user and admin with AR experience.
* As technology evolves and makes AR relevant it will change the way we live our lives. Web stores will embrace 3D as a shopping experience, making it easier to browse and shop. The cons to the technology, which can give you the advantage you need to fight back against these well-funded tech innovations.

BIBLOGRAPHY

1. **Books**

* Kenneth.S.Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, First Edition
* Allen B. Downey Think Python: An Introduction to Software Design, Second Edition
* **OpenCV (2019)** OpenCv—Python Core— [online]

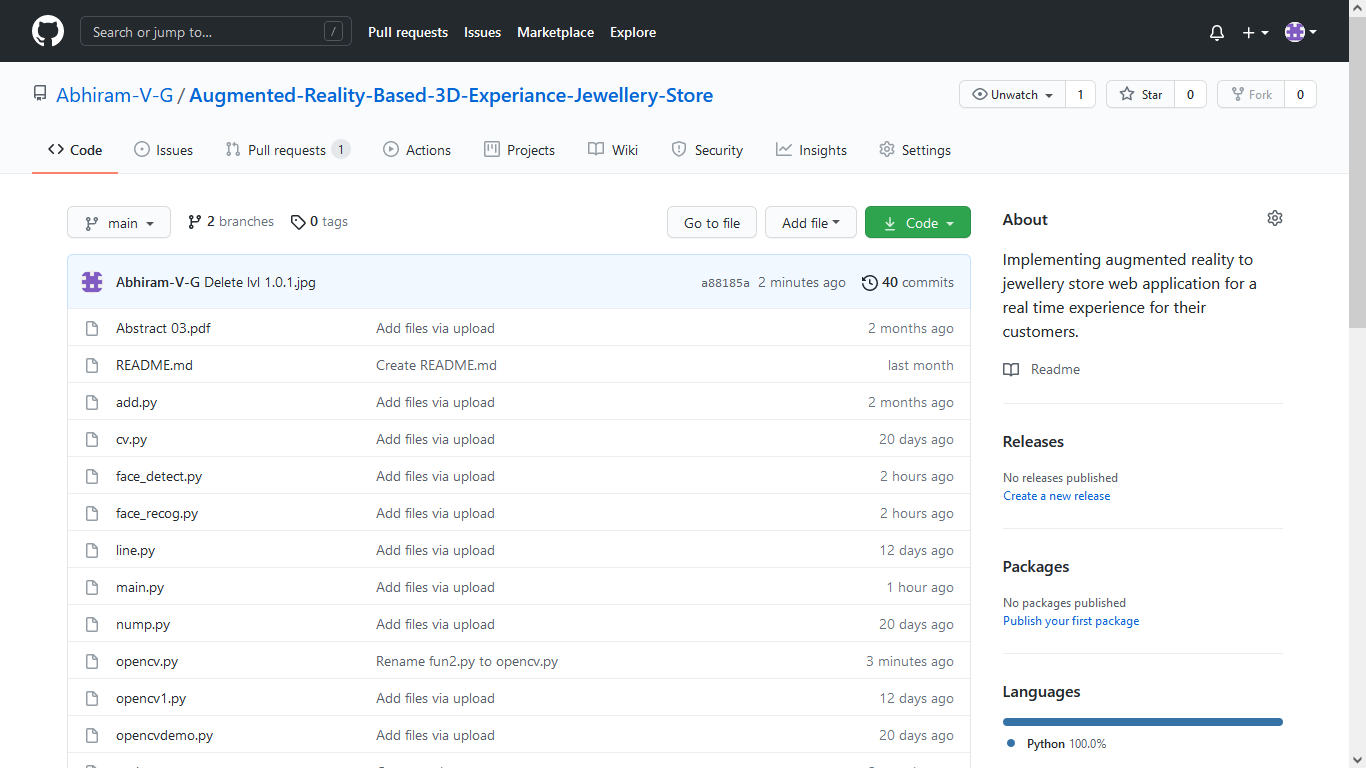
1. **Websites**

* <https://www.javatpoint.com/python-tutorial>
* <https://www.tutorialspoint.com/python/index.html>
* <https://www.tutorialspoint.com/mysql/mysql-introduction.html>
* <https://www.tutorialspoint.com/opencv/opencv_overview.htm>

1. **Journal and Publications**

* Jacob Ziollner, Erick Mendez and Daniel Wagner, “Augmented Reality Remote Collaboration with Dense Reconstruction”, DAQRI Vienna [2018]
* Rantala, I., A. Colley, and J. Häkkilä. Smart Jewelry: Augmenting Traditional Wearable Self-Expression Displays in Proceedings of the 7th ACM International Symposium on Pervasive Displays. 2018. ACM.

Git History



APPENDICES

1. LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| Table No | Table name | Page No |
| 5.1 | Admin | 34 |
| 5.2 | Customer\_registration | 34 |
| 5.3 | Customer\_login\_id | 35 |
| 5.4 | Product | 35 |
| 5.5 | Product\_details | 36 |
| 5.6 | Product\_keywords | 37 |
| 5.7 | Category | 37 |
| 5.8 | Order | 38 |
| 5.9 | Order\_details | 38 |
| 5.10 | Order\_update | 39 |
| 5.11 | Customer\_feedback | 39 |
| 5.12 | Vendor | 40 |
| 5.13 | Inventory | 41 |
| 5.14 | Inventory\_products | 41 |
| 5.15 | Input and output | 44 |
| 7.1 | Testing Admin login | 53 |
| 7.2 | Testing Customer login id | 54 |
| 7.3 | Testing Add Product | 55 |
| 7.4 | Testing Add Vendor | 57 |
| 7.5 | Testing Update Stock | 58 |

1. LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Figure No | Figure name | Page no |
| 5.1 | DFD Components | 23 |
| 5.2 | Context diagram (level 0) | 24 |
| 5.3 | Admin login (level 1) | 25 |
| 5.4 | User login (level 1) | 26 |
| 5.5 | Admin login (level 1.0) | 27 |
| 5.6 | Product Management(level 1.0) | 28 |
| 5.7 | Order management (level 1.0) | 29 |
| 5.8 | Inventory (level 1.0) | 30 |
| 5.9 | Vendor management | 31 |
| 5.10 | User | 32 |
| I. | Admin login | 63 |
| II. | Add Category | 63 |
| III. | Add Product | 64 |
| IV. | Upload pictures | 65 |
| V. | Add Vendor | 65 |
| VI. | Edit Vendor | 66 |
| VII. | Update stock | 66 |
| VIII. | View update | 67 |
| IX. | View orders | 67 |
| X. | Order update | 68 |
| XI. | Dash board 1 | 68 |
| XII. | Dash board 2 | 69 |
| XIII. | Dash board 3 | 69 |
| XIV. | User sign-in | 70 |
| XV. | User login | 70 |
| XVI. | Product View | 71 |
| XVII. | Preview of product in AR | 71 |

1. ABBREVIATIONS AND NOTATION

* DFD

(Data Flow Diagram) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

* DB

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.