**INTRODUCTION**

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming common place. An online store is a virtual store on the Internet where customers can browse the catalogue and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

Customizable dress purchasing site is a web application for online dress purchasing which has some advanced features. In this project customers can customize or design their products. Customers can chat with the designers and they can give suggestions for customer’s needs. Customers can order dress after they are fully satisfied with the product. In this application a chat session is provided for customers and design. Admin can assign the received orders to tailors for completing the alterations required for the customers.

**REQUIREMENT ANALYSIS AND SPECIFICATION**

**REQUIREMENT ANALYSIS**

Requirement analysis is significant and essential activity after elicitation .we analyses, refine and scrutinize the gathered requirement to make consistent and unambiguous requirements. This activity reviews all requirements and may provide a graphical view of the entire system. After the completion of the analysis, it is expected that the understandability of the project may improve significantly. Here, we may also use the interaction with the customer to clarify points of confusion and to understand which requirements are more important than others.

One effective way to find out what the customer wants is to construct a prototype, something that looks and preferably acts us part of the system they want. We can use their feedback to modify the prototype until the customer is satisfied continuously. Hence, the prototype helps the client to visualize the proposed system and increase the understanding of requirements. In requirement analysis the graphical view may help to find incorrect, inconsistent and missing requirements. Such model includes the Dataflow diagram, Entity Relationship Diagram, etc.

After modeling the requirements, we will have a better understanding of the system behavior. The inconsistencies and ambiguities have been identified and corrected. Elicitation and analysis activities have provided better insight into the system. After the analyses requirements and the next step is to document these requirements in a prescribed format.

**EXISTING SYSTEM**

There are many online shopping sites, but most of them share the standard features of online shopping. We can only buy the products in the site as it is. We can only customize or alter the dress after buying the product. There are many situations when we need to alter the dress or need to customize that. In the existing online purchasing sites we can’t do this. Currently there are no online shopping sites that provide customization of products as required by customers. They only provide dresses in some fixed sizes and designs. If we need any changes customers can do that only after busying the products.

**PROPOSED SYSTEM**

The proposed system is entirely concentrating on denying these kinds of unhappy things happening to any customer. And the ultimate goal is the satisfaction of customer that, what he is buying is exactly what he looked for. This is made possible by making a space for them to customize what is seen or they can entirely design a dress from the scratch. The system assists them in their whole designing process by providing similar designs, patterns or even relevant suggestions at each phase of designing through live chats. They can also give their size specimens and any other requirements if any. The order is placed only after the customer is fully satisfied with what they see. The admin then assigns the order to any tailor in his company and also give the payment and delivering details to the customer.

**MODULES**

* Admin
* Designer
* Tailors
* Customer

**ADMIN**

Admin plays an important role in this system. He must have an effective communication with both customer and tailors who are not connected to each other. that is, admin is the interface between customer and tailor. He accepts the order customized and placed by customer and assigns to any tailor under his company and have frequent updates from tailor about the item. Admin can manage tailors and products, assign works to tailors, view complaints or feedback from customers and manage tailor leave applications.

**DESIGNER**

Designer gives design suggestion to the customer. He can also give suggestions by analyzing the physical features of customer like height, skin color, etc. Designer can chat with customers and give suggestions. They upload a final design of the outfit if the customer is satisfied with the design.

**TAILOR**

Tailors will receive works and can accept or reject the work assigned by the admin. They can regularly update about the accepted works and view complaints and feedbacks from customers. Tailors can also apply for leave and view leave status.

**CUSTOMER**

Customer interacts with admin and designer. Customers customize the product they want or can design the dress from scratch. If needed, in designing phase customer can be accompanied with expert advice from the designer. He can also simply upload photos of any dress to convey his idea and can also just say any changes he needed with the dress in that photo he uploaded. Customer can order dress and view order status. They can send feedbacks and complaints. Customers can also chat with designers and view suggestions from designers.

**REQUIREMENT SPECIFICATION**

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and nonfunctional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of software system. To achieve this we need to continuous communications with customers to gather all requirements.

A good SRS defines the how software system will interact with all internal modules ,hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the SRS document on QA lead managers creates test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results. It is highly recommended to review or test SRS document before start writing test cases and making any plan for testing.

**Functional Requirements**

* Functional Requirements defines a function of a system or its component, Where a function is described as a specification of behavior between outputs and inputs.
* Functional Requirements may involve calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. It specifies a particular result of a system.

**Nonfunctional Requirements**

* A nonfunctional requirement defines the quality attribute of a software system.
* They judge the software system based on reliability, usability, security, portability and other nonfunctional standards that are critical to success of the software system.
* The process of specifying nonfunctional requirements requires the knowledge of the functionality of the system as well as the knowledge of the context within which the system will operate.
* Security: The System should be secure, that the details of users are kept confidential
* Reliable: The System should be reliable on the information provide
* User friendly :The system should be simple and usability
* Platform independent :The software used here is platform independent
* Efficiency :The system should be able to operate faster than the existing system
* Performance: The system provide good performance

**FEASIBILITY STUDY**

A feasibility study is a preliminary study undertaken to determine and document a project‟s viability. The results of this study are used to make a decision whether to proceed with the project. If it indeed leads to a project being approved, it will – before the real work of proposed project starts – be used to ascertain the likelihood of the project‟s success. It is an analysis of possible alternative solutions to a problem and recommendation on the best alternative. It, for example, can decide.

Whether an order processing be carried out by a new system more efficiently than the previous one. The feasibility study proposes one or more conceptual solutions to the problem set for the project. The conceptual solution gives an idea of what the new system will look like. They define what will be done on the computer and what will remain manual. It also indicates what input will be needed by the system and what outputs will be produced. These solutions should be proven feasible and a preferred solution is accepted.

The feasibility study environment enables all alternatives to be discussed and evaluated. This phase starts with an identification of the main characteristics of the required system. During this stage it is important to collect information as much as possible about the software package that might meet the specification from as many sources as possible. Normally, the central endeavor of a feasibility study is a cost benefit analysis of various alternatives. It can be defined as a systematic comparison between the cost of carrying out a service or activity and the value of that service or activity. The main benefits are qualitative than quantitative.

A feasibility study could be used to test a new working system, which could be used because:

* The current system may no longer suit its purpose,
* Technological advancement may have rendered the current system obsolete,
* The business is expanding, allowing it to cope with extra workload,
* Customers are complaining about the speed and quality of work the business provides.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were:

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility

The requirements of the system are specified with a set of constraints such as system objectives and the description of the out puts. Three key factors are to be considered during the feasibility study.

**OPERAIONAL FEASIBILITY**

The proposed project is operationally feasible. Because the proposed python application is easily operated by the users. The proposed system is operated by the admin and can use by the users. The user who have basic

internet knowledge, he can use the web site for registration and then use the website for their intended applications. For operating the web content, the users need only the basic internet knowledge and an internet connection. The kisanseva can be easily accessible and operated by the users who has a device and internet connection. Now a day almost persons have smartphones. Hence we can be say that the proposed system is operationaly feasible.

**TECHNICAL FEASIBILITY**

The proposed system meets all the requirement of the Technical feasibility. Because the implementation of the project needs no technological difficult. The web part of the project can be easily implemented in a user friendly IDE, that is in Pycharm. The system can be easily developed by using the current technologies. The web application is developed in the most popular interpreted, high-level and general-purpose programming language PYTHON . So the development of the project does not meet any type of the technical difficult. Hence we can say that the proposed system is technically feasible.

**ECONOMIC FEASIBILITY**

The proposed project is economically feasible. Because once the system is put into its use in the current market the system provides economical advantage to the firm. Also the firm can afford the cost to implement the project.

The proposed project car crash provides tangible and intangible benefits comparing to the existing projects. This system does not need any initial investments and it can improve the quality of service.

**PROJECT PLANNING AND SCHEDULING**

Planning and scheduling are distinct but inseparable aspects of managing the successful project .The process of planning primarily deals with selecting the appropriate procedures in order to achieve the objectives of the project. Scheduling converts the project action plans for scope, time cost and quality into an operating timetable.

Project planning is concerned with identifying the activities, milestones, and deliverables, produced by project. A plan must then be drawn up to guide the development towards the project goals. A plan drawn up at the start of a project should be used as the driver for the project. Those initial plans is not static but must be modified as the project progresses and better information becomes available. Planning is required for development activities from specification through delivery of the system.

Project scheduling is a mechanism to communicate what tasks need to get done and which organizational resources will be allocated to complete those tasks in what timeframe. A project schedule is a document collecting all the work needed to deliver the project on time. Without a full and complete schedule, the project manager will be unable to communicate the complete effort in terms of cost and resources, necessary to deliver the project.

**SYSTEM SPECIFICATION**

System environment means the essential features for the working of our new application software.

System configuration is classified into three:

* Hardware Specification
* Software Specification
* Tools

**HARDWARE SPECIFICATION**

* Processor : Intel core i3 or above
* Clock Speed : 500 MHZ
* System Bus : 32Bit
* RAM : Minimum 2GB(recommended 8GB)
* HDD : 20GB
* Monitor : SVGA COLOR
* Key Board : 108 Keys
* Mouse : Any type of mouse

**Software Requirement**

* Operating System : Windows 10/XP OR Linux
* Coding Language : python Django
* Browser : Mozilla Firefox/ Google Chrome /Internet Explorer
* IDE : Anaconda
* Protocol : HTTP, SMTP.

**TOOLS**

**CSS**

It is a style sheet language used for describing the presentation of a Document written in a markup language.

**MySQL**

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client– server database engine. Rather, it is embedded into the end program.

**WINDOWS 7/8/10**

Windows 7/8/10 is a personal computer operating system developed and released by Microsoft as part of the Windows NT family of operating system.

**PYTHON DJANGO**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

**PYTHON**

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbagecollected. Python is often described as a "batteries included" language due to its comprehensive standard library.

**HTML**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

**JAVA SCRIPT**

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based objectorientation, and first-class functions.

**SYSTEM DESIGN**

**SYSTEM DESIGN**

System Design deals with transforming the customer requirements, into a form that is implemented using a programming language. It is the phase in which, the detailed design of the system is accomplished and the user oriented performance specification converted into a technical design specification.

During this phase, decision about overall architecture was taken. The system organized into subsystem based on the analysis structure. A personal computer with ordinary specification was chosen for the implementation of the design. The major activities include general system design, design of all outputs, input design and the design of the database.

Principle activities performed during this phase include,

• Input Design

Input design is the process of converting the user originated inputs to a computer format. The input design involves determining what the inputs are, how the data should be performed, how to validate data, how to minimize data entry and how to provide a multiuser facility. The design for handling input specifies how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention and if includes specifying the means by which actions are taken. A system user interacting through a system must be able to tell the system whether to accept input produce a report or end processing. The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information extreme care is taken to obtain the information. If the data going into the system is incorrect then processing and outputs will magnify this error. All input data are validated in the order and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, then it is transferred to the appropriate tables in the database.

We have to keep in mind the following things to design the system

♣What data to input?

♣What medium to use?

♣The dialogue to guide users in providing input

♣Methods for performing input validation and steps to follow when errors occur Input requirement gathering was one of the major trivial process in web or android application development. The project involves text inputs.The inputs can be entered through keyboard and mouse. The text input is gathered by forms with text boxes.

• Database Design

• Code design

• Output design

**Partition requirement**

The requirements are analysed and collected into related group

**Identify subsystem**

Different subsystems that can individually or collectively meet the requirements are identified

**Assign requirements to subsystem**

The requirements are assigned to subsystem

**Specify subsystem**

The specific functions provided by each subsystem are specified

**Define subsystem interface**

This involves defining the interface that are provided and required by each subsystem. Once these interface have been agreed parallel development of the subsystem become possible

**DATA FLOW DIAGRAM**

A data flow diagram (DFD) or bubble chart is a graphical tool for structured analysis. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates output data flows which go to other processes or external entities or files. Data in files may also flow to processes as input. DFDs can be hierarchically organized, which help in partitioning and analyzing large systems. As a first step, one data flow diagram can depict an entire system which gives the overview. It is called as context diagram of level 0DFD. The context diagram can be further expanded. The successive expansion of a DFD from the context diagram to those giving more details is known as leveling of DFD. Thus a top down approach is used, starting with an overview and then working out the details. The main merit of DFD is that it can provide an overview of what data a system would process, what transformation data are done, what files are used, and where the results flow. The data flow has been represented as a hierarchical DFD. Context level DFD is drawn first then the processes were decomposed into several elementary levels and were represented in the order of importance. The data flow diagram (DFD) is one of the most important tools used by system analyst. Data flow diagrams are made up of a number of symbols, which represent system components. Most data flow modeling methods use four kinds of symbols. These symbols are used to represent four kinds of components. Processes, data source, data flows and external entities.

**BASIC DFD SYMBOLS**

• **ENTITY**

A source or sink is a person or part of an organization which enter or receives information from the system, but is considered to be outside the contest of data flow model.

**• PROCESS**

A circle stands for process that converts data into information. A process represents transformation where incoming data flows. A data flow is a route, which enables are changed into outgoing data flows

• **DATABASE**

A data store (database) is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational databases. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double headed arrow is used.

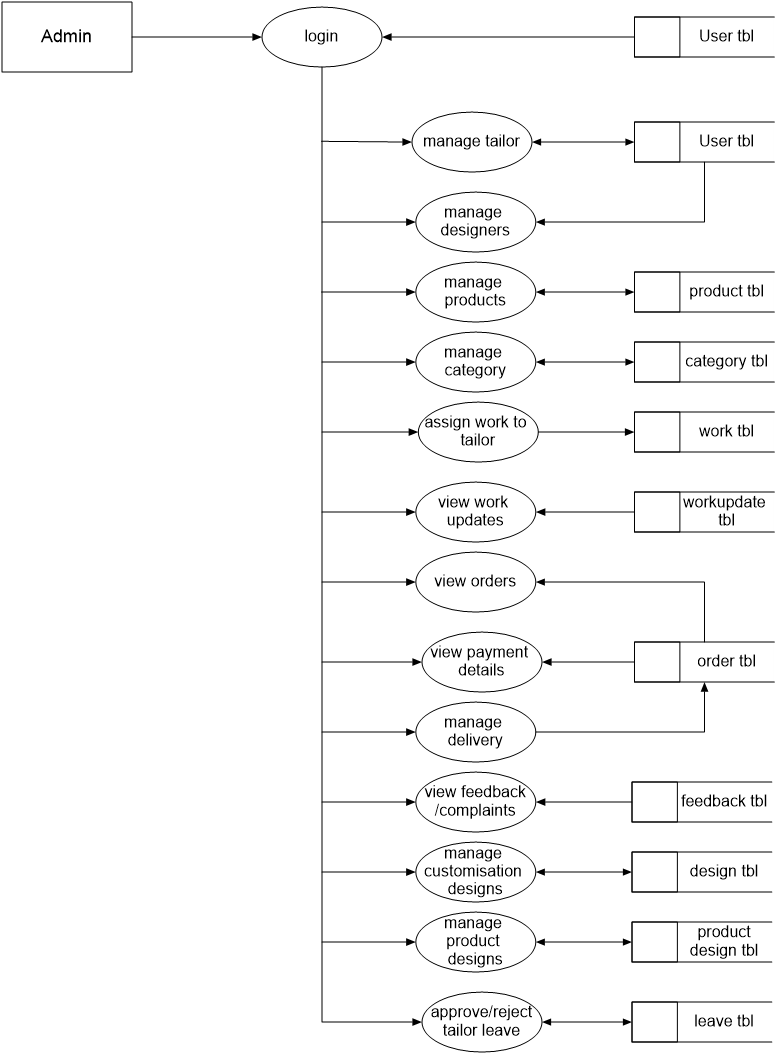
**DATA FLOW(Arrow)**

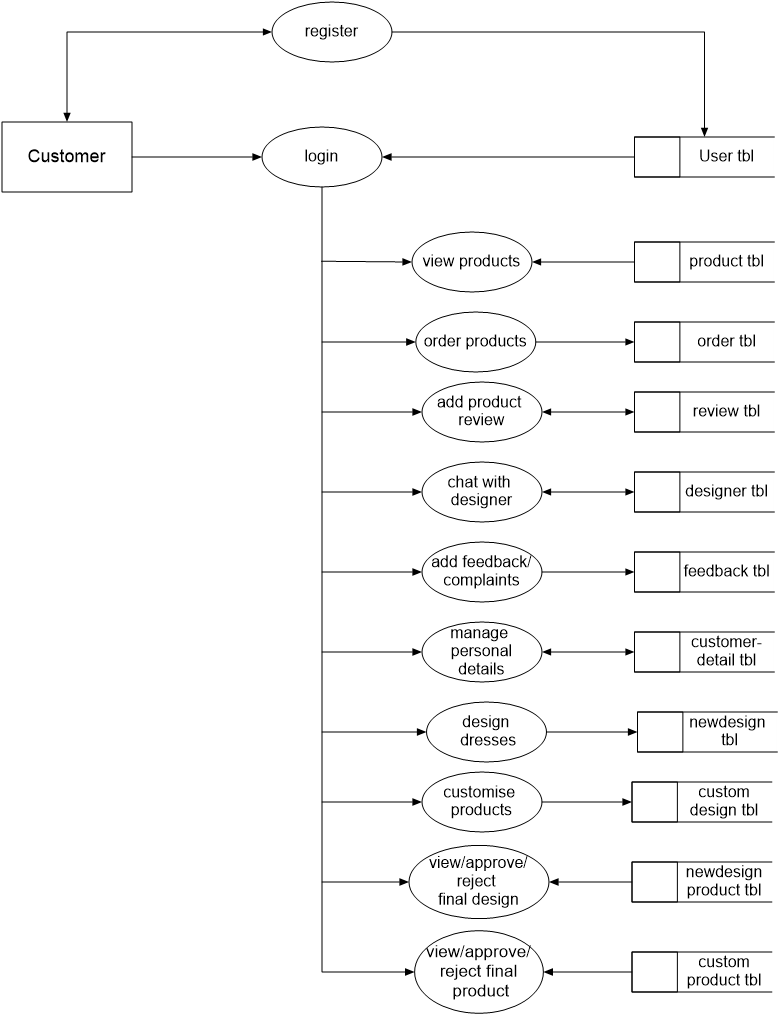
A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.

**LEVEL 0**

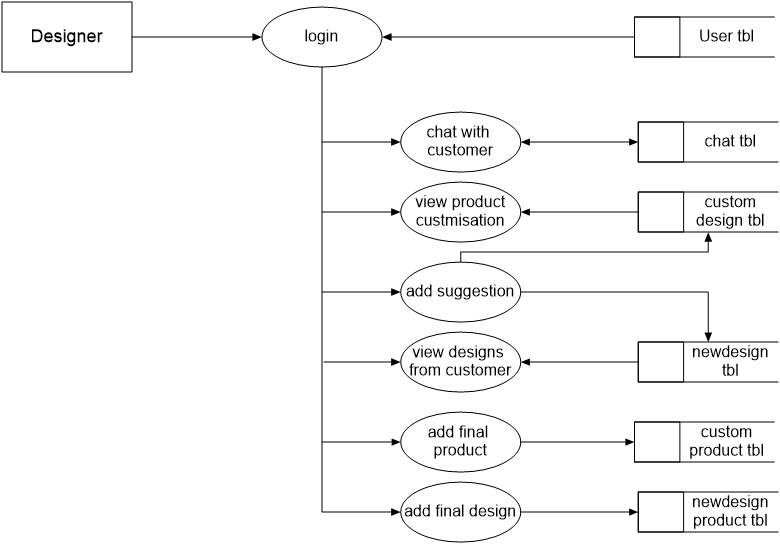


**ADMIN LEVEL 1**

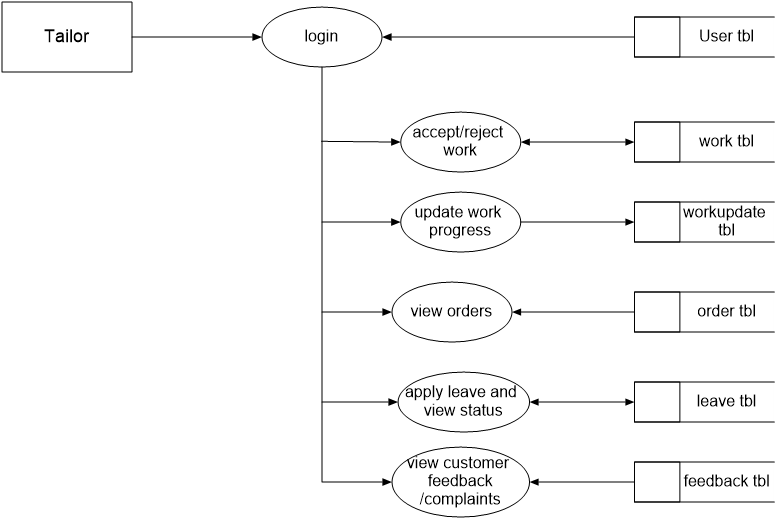
****

**CUSTOMERLEVEL1**

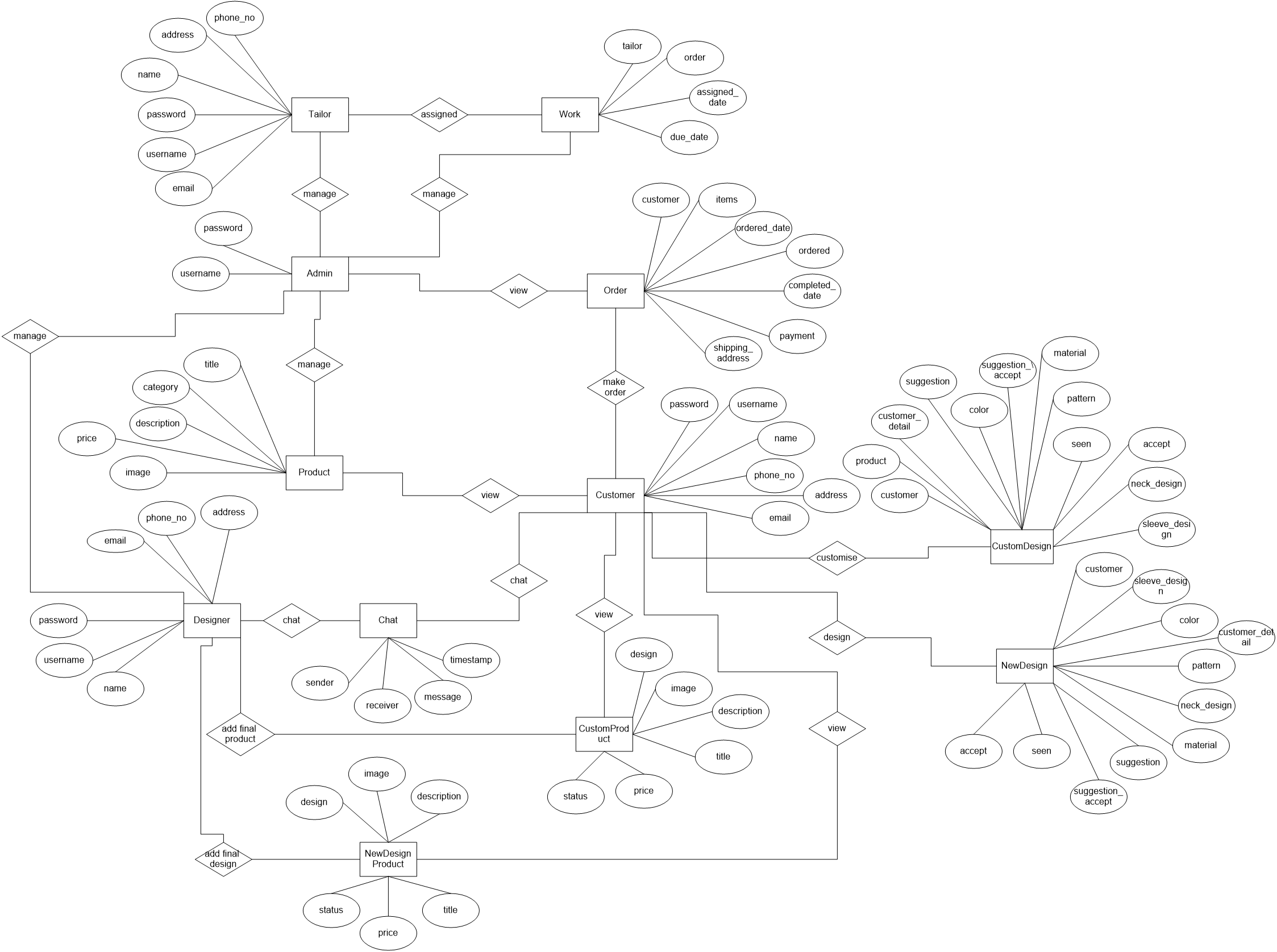
**DESIGNER LEVEL1**

****

**TAILOR LEVEL1**

****

**ER DIAGRAM**

****

### DATABASE DESIGN

Database design manages large bodies of information. Database is the collection of related data. It provides safety of information. A database is a collection of inter-related data stored with minimum redundancy to save many users quickly and effectively. Database runs parallel without application design.

### Data Normalization

The normalization simplifies the entries, removing redundancies from the system data and finally builds a data structure, which is both flexible and adaptable the system. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

The different normal forms applied during the design of the data base are:

### First Normal Form:

A relationship is said to be in first normal form if and only if it satisfies the constraints that it contains atomic values. It states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. It was defined to disallow multivalued attributes, composite attributes, and their combinations.

First normalization used in all tables in the database.

### Second Normal Form:

A relationship is said to be in second normal form if it satisfies the 1NF condition for the primary key and every non primary key attributes of the relation should not depend on the primary key alone. Second normal form (2NF) is based on the concept of full functional dependency. A functional dependency X - Y is a fully functional dependency if removal of any attribute A from X means that the dependency does not hold anymore.

This normalization can be used in all the tables in the database because all the table contain a primary key.

∙ Third Normal Form:

A relationship is said to be in third normal form if and only if satisfies all the 2NF conditions and non-key attributes of the relation should not depend on other non-key attributes. Third normal form (3NF) is based on the concept of transitive dependency. A functional dependency X ~ Y in a relation schema R is a transitive dependency if there is a

set of attributes Z that is neither a candidate key nor a subset of any key of R, and both X -Z and Z –Y hold.

**Table design**

User table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| user\_id | int | unique,primarykey |
| username | varchar |  |
| password | varchar |  |
| name | varchar |  |
| phone\_no | varchar |  |
| email | varchar |  |
| address | varchar |  |

DressCategory table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| category\_id | int | Primary key |
| category | varchar |  |
| image | file |  |

Product table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey,unique |
| title | varchar |  |
| price | float |  |
| category | int | foreignkey key |
| description | varchar |  |
| image | file |  |

Review table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey,unique |
| user\_id | int | foreignkey key |
| item | Int | foreignkey key |
| rating | Varchar |  |
| date\_added | Date |  |

Design table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primary key , unique |
| category | int | foreignkey key |
| type | varchar |  |
| name | varchar |  |
| image | file |  |
| orders | int |  |

ProductDesign table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primary key , unique |
| product | int | foreignkey key |
| type | varchar |  |
| name | varchar |  |
| image | file |  |
| orders | int |  |

NewDesign table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primary key , unique |
| customer | int | Foreignkey |
| customer\_detail | int | foreignkey |
| color | int | foreignkey |
| pattern | int | foreignkey |
| category | int | foreignkey |
| material | int | foreignkey |
| neck\_design | int | foreignkey |
| sleeve\_design | int | foreignkey |
| complete | int |  |
| suggestion | varchar |  |
| suggestion\_accept | int |  |
| accept | int |  |
| seen | bool |  |

CustomDesign table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primary key , unique |
| customer | int | Foreignkey |
| product | int | Foreignkey |
| customer\_detail | int | foreignkey |
| color | int | foreignkey |
| pattern | int | foreignkey |
| category | int | foreignkey |
| material | int | foreignkey |
| neck\_design | int | foreignkey |
| sleeve\_design | int | foreignkey |
| complete | int |  |
| suggestion | varchar |  |
| suggestion\_accept | int |  |
| accept | int |  |
| seen | bool |  |

CustomProduct table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primary key , unique |
| design | int | Foreignkey |
| title | varchar |  |
| description | varchar |  |
| price | float |  |
| image | file |  |
| status | int |  |

NewDesignProduct table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primary key , unique |
| design | int | Foreignkey |
| title | varchar |  |
| description | varchar |  |
| price | float |  |
| image | file |  |
| status | int |  |

CustomOrderItem table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| Order\_item\_id | int | Primary key , unique |
| customer | int | Foreignkey |
| ordered | bool |  |
| item | int | foreignkey |
| quantity | int |  |

DesignOrderItem table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| Order\_item\_id | int | Primary key , unique |
| customer | int | Foreignkey |
| ordered | bool |  |
| item | int | foreignkey |
| quantity | int |  |

Order Item table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| Order\_item\_id | int | Primary key , unique |
| customer | int | Foreignkey |
| ordered | bool |  |
| item | int | foreignkey |
| quantity | int |  |

Shipping Address table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| customer | int | foreignkey |
| address | varchar |  |
| email | varchar |  |
| phone\_no | varchar |  |
| city | varchar |  |
| state | varchar |  |
| pincode | varchar |  |

Payment table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| customer | int | foreignkey |
| amount | varchar |  |
| card\_no | varchar |  |
| card\_cvv | varchar |  |
| expiry\_month | varchar |  |
| expiry\_year | varchar |  |

Order table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| customer | int | foreignkey |
| items | int | foreignkey |
| custom\_product | int | foreignkey |
| design\_product | int | foreignkey |
| ordered\_date | date |  |
| ordered | bool |  |
| shipping\_address | int | foreignkey |
| payment | int | foreignkey |
| completed | bool |  |
| Completed\_date | date |  |

Feedback table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| customer | int | foreignkey |
| subject | varchar |  |
| feedback | varchar |  |
| date | varchar |  |

TailorLeave table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| tailor | int | foreignkey |
| From\_date | varchar |  |
| To\_date | varchar |  |
| reason | varchar |  |
| Submitted\_on | varchar |  |
| reason | varchar |  |

Chat table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| sender | int | foreignkey |
| receiver | int | foreignkey |
| message | varchar |  |
| image | file |  |
| timestamp | varchar |  |
| c\_seen |  |  |
| d\_seen |  |  |

CustomerDetail table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| customer | int | foreignkey |
| height | varchar |  |
| size | varchar |  |
| Skin\_color | varchar |  |

Work table

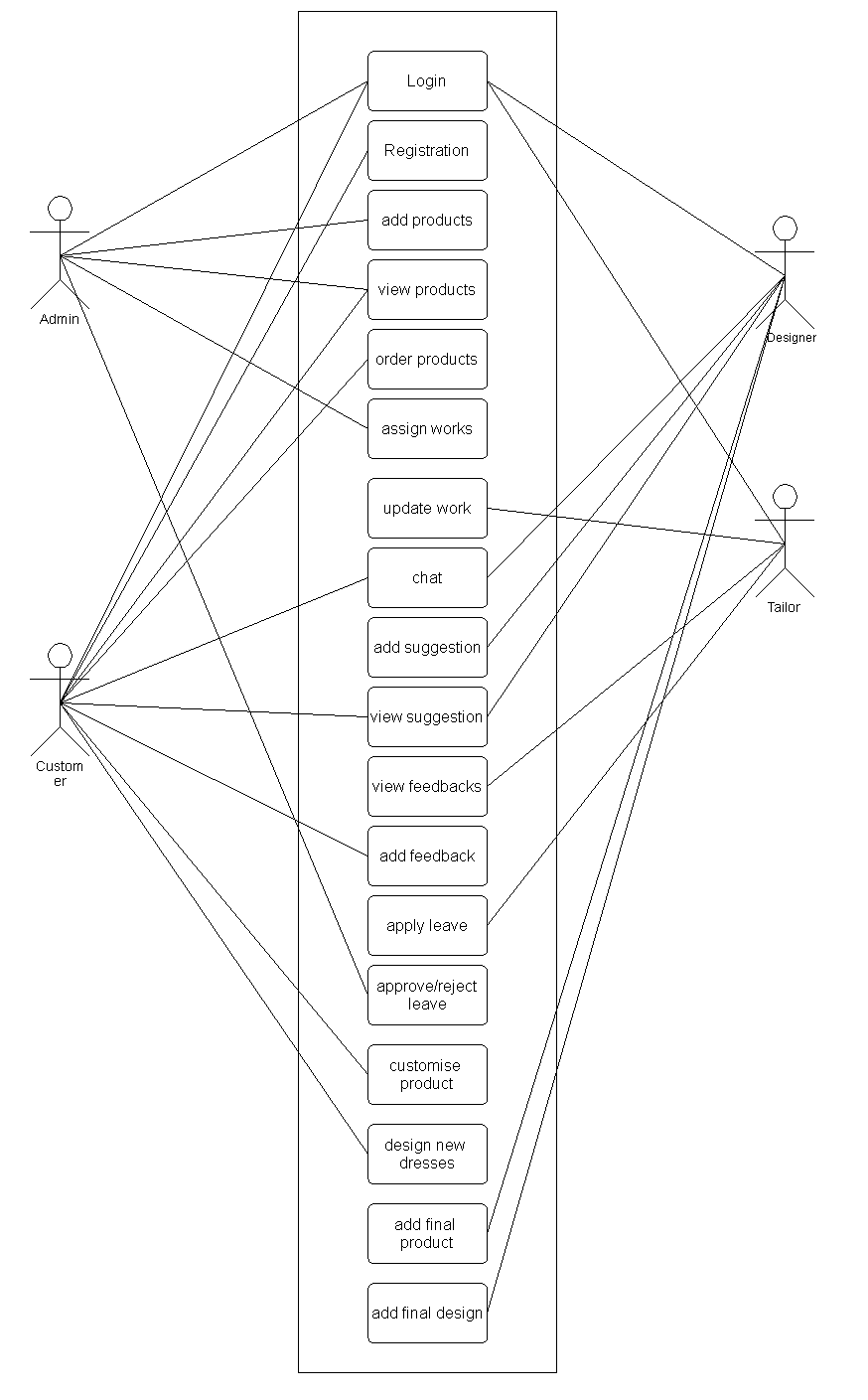
|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| tailor | int | foreignkey |
| order | int | foreignkey |
| Assigned\_date | varchar |  |
| Due\_date | varchar |  |

WorkUpdate table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Attribute** |
| id | int | Primarykey |
| work | int | foreignkey |
| date | varchar |  |
| update | varchar |  |

**USECASE DIAGRAM**

Use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a “system” is something being developed or operated. The “actors” are people or entities operating under defined roles within the system. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration. They provide a good high level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.



• **OUTPUT DESIGN**

Effective output design will improve the clarity and performance of output. Output design phase of the system is concerned with the convergence of information‟s to the end user friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and thereby enhancing the process of decision making.

The Outputs from the SYSTEM ANALYSIS app consist of different formats. It provides search details in text format. The most attractive feature of this application is it shows the results of search contents after efficient filtering and pruning techniques.Efficient and eligible output design should improve the system‟s relationship with the user and help in decision making.

Outputs are the most important and direct source of information to the user and to the management. Output design generally deals with the results generated by the system. The output data is in the format of text.

**CODE DESIGN**

The goal of the coding is to translate the design of the coding into a given programming language, Coding phase affect both the testing and maintenance phase, Better codes will reduce the testing and maintenance costs. For a given design the aim of the coding is to implement the design in the best possible manner. Simplicity and clarity are main properties of the coding. This is one of the operational phase in the system development life cycle. This phase deal with coding of the proposed system using the development tools. The coding depends on the logical and operational model of the proposed system.

**SYSTEM DEVELOPMENT**

**SYSTEM DEVELOPMENT**

The goal of the development stage is to translate the design of the system produced during the design phase into code in a given programming language which can be executed by a computer and which performs the computations specified by the design.

Here we are using python with HTML .It is intended for the purpose of the to create a user friendly interface for the website. Having a good user interface make it easier for the user to use and understand the different functionalities of the website .

For the intended project to work on, we need to implement its required hardware and software requirements .This system is build using python and HTML using SQLite based on Windows operating System. Memory and Hard disk should confirm according to hardware specification mentioned above.

Here i used Front-end is python, HTML .The backend is SQLite. The proposed system can be build using front-end and back-end languages.

**SYSTEM TESTING**

**SYSTEM TESTING**

System testing is the stage of implementation highly aimed at ensuring that the system works accurately and efficiently before the live operation commences. Testing is vital to the success of the system. The primary objective of testing is to derive a set of tests that has the highest likelihood for uncovering defects in the software. The system test in implementation should conform that all is correct and an opportunity to show the users that the system work as expected. It accounts the largest percentage of technical effort in the software development process. Testing phase in the development cycle validates the code against the functional specification. The performance of the system is measured in this phase. Testing is a set activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it is vital success of the system. The testing can be a set of verification and validation process. Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

**UNIT TESTING**

Unit testing focuses verification efforts on the smallest unit of software design module. To check whether each module in this software works properly so that it gives desired outputs to the given inputs. All validations and conditions are tested in the module level in the unit test control parts are tested to ensure the information correctly flows into, and out of the program unit under test. Boundary conditions are tested to ensure that modules operate at the boundaries. All independents parts through the control structure ensure that all statements in a module have been executed at least once. We test all the modules of the SYSTEM ANALYSIS app and works properly.

**INTEGRATION TESTING**

Data can be lost across an interface, one module have an adverse effect on the other sub-functions, when combined may not produce the desired functions. Integrated testing is the systematic to uncover the errors within the interface. This testing is done by inputting necessary values and data in the sequential order. As data of one unit is needed for working of other. The need for integrated system is to find the overall system performance.

**Project Aspect:**

After connecting the back-end and the front-end as whole module, the data entered in the front-end once submitted were successfully entered in the data base. On request, data were successfully retrieved in to forms.

**SYSTEM TESTING**

System testing is an ongoing activity throughout the project. The logical design and the physical design should be thoroughly and continually examined on paper to ensure that they will work when implemented. Thus the system test in implication should be a confirmation that all is correct and an opportunity to show the user that the system works. This checks all the elements of the system. The most important elements are hardware, software, people and information. System testing is either conducted by the programmer, the programmer‟s manage or by a Quality Control Unit within the IT organization. The goal is to ensure that all of the code works as defined by the requirements and design documents before it is delivered to the Users.

**Project Aspect:**

After the development system, we checked the overall system works correctly as to ensure that code works as defined.

**WHITE BOX TESTING**

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

**Project Aspect:**

We have checked the flow of control, branching and looping.

**VALIDATION TESTING**

Validation test is defined with a simple definition that validation succeeds when the system functions in a manner that can be reasonably accepted by the customer. Validation is done to see whether the corresponding entries made in the tables are correct. Proper validations are done in case of insertion and updating of tables. If any such arises, then proper error messages or warning, if any, has to be displayed.

In SYSTEM ANALYSIS app the validation testing is done to different fields especially in registration and signup field. Where the validation done through java program.Main validation used is NULL value in the text boxes.And also done Mobile number , E-mail validation.

**Project Aspect:**

Check name, mail id, and password correctly.

**SYSTEM IMPLEMENTATION AND MAINTANANCE**

**SYSTEM IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and it constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning, major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation. An implementation coordinating committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

**IMPLEMENTATION METHODS**

Implementation is the final and important phase. The most critical stage in achieving a successful new system and in giving the users confidence that the new systems will work and be effectives. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system. The implementation process begins with preparing the plan for implementation of the system. Once the planning has been completed, the major effort in the computer department is to ensure that the programs in the system are working properly. At the same time the HR department must concentrate on training user staff.

**SYSTEM MAINTANANCE**

the results obtained from the evaluation process help the organization to determine whether its information systems are effective and efficient or otherwise. The process of monitoring, evaluating, and modifying of existing information systems to make required or desirable improvements may be termed as System Maintenance. System maintenance is an ongoing activity, which covers a wide variety of activities, including removing program and design errors, updating documentation and test data and updating user support. For the purpose of convenience, maintenance may be categorized as:

• Corrective

• Adaptive

• Perfective

• Preventive

**Corrective Maintenance**

This type of maintenance implies removing errors in a program, which might have crept in the system due to faulty design or wrong assumptions. Thus, in corrective maintenance, processing or performance failures are repaired.

**Adaptive Maintenance**

In adaptive maintenance, program functions are changed to enable the information system to satisfy the information needs of the user. This type of maintenance may become necessary because of organizational changes which may include:

• Changes in organizational procedures

• Changes in organizational objectives ,goals ,policies ,etc.,

• Changes in forms

• Changes in information needs of managers

• Changes in system controls and security needs, etc.

**Perfective Maintenance**

Perfective maintenance means adding new programs or modifying the existing programs to enhance the performance of the information system. This type of maintenance undertaken to respond to user‟s additional needs which may be due to the changes within or outside of the organization. Outside changes are primarily environmental changes, which may in the absence of system maintenance; render the information system ineffective and inefficient. These environmental changes include:

• Changes in governmental policies, laws, etc.,

• Economic and competitive conditions, and New technology

**Preventive Maintenance**

It involves performing activities to prevent the occurrence of errors. It reduces the software complexity thereby improving program understandability and increasing software maintainability.

**FUTURE ENHANCEMENT**

**FUTURE ENHANCEMENT**

CUSTOMISABLE DRESS PURCHASING SITE is a web application has been designed and developed according to the current techniques and scopes of designing and development tools. This system is very flexible so that the maintenance and further amendments based on the changing environment and requirements can be made easily with adding further information. Further enhancement is possible to updating evaluating tools. This can be restructured as required.

• Android applications can be developed for the easiness of user

.

# *CONCLUSION*

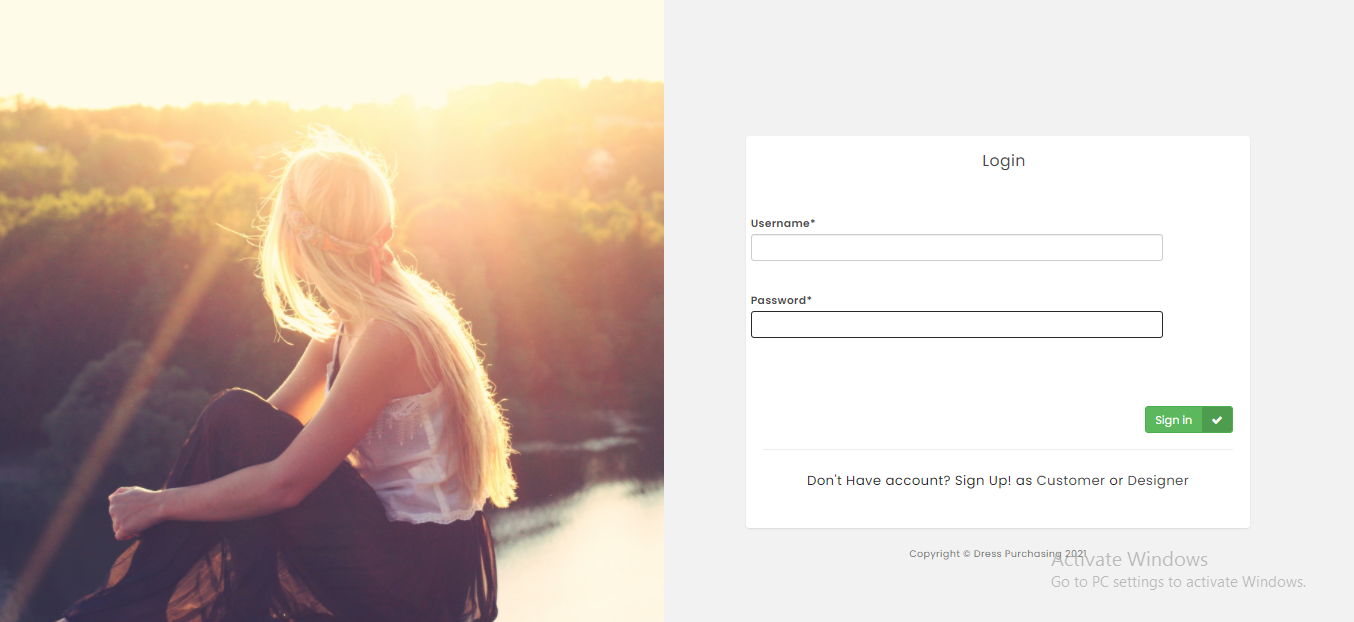
**11. CONCLUSION**

The project named CUSSTOMISABLE DRESS PURCHASING SITE developed as a web application for online shopping. This application provides users the flexibility to customize their products or design products. Customers can order products from the site and also can chat with the designers. They can ask suggestion to the designers through chatting. Admin manages the overall working of the application. Admin can manage products and can assign the orders received to tailors. Designers can chat with the customers and suggest products for customers.

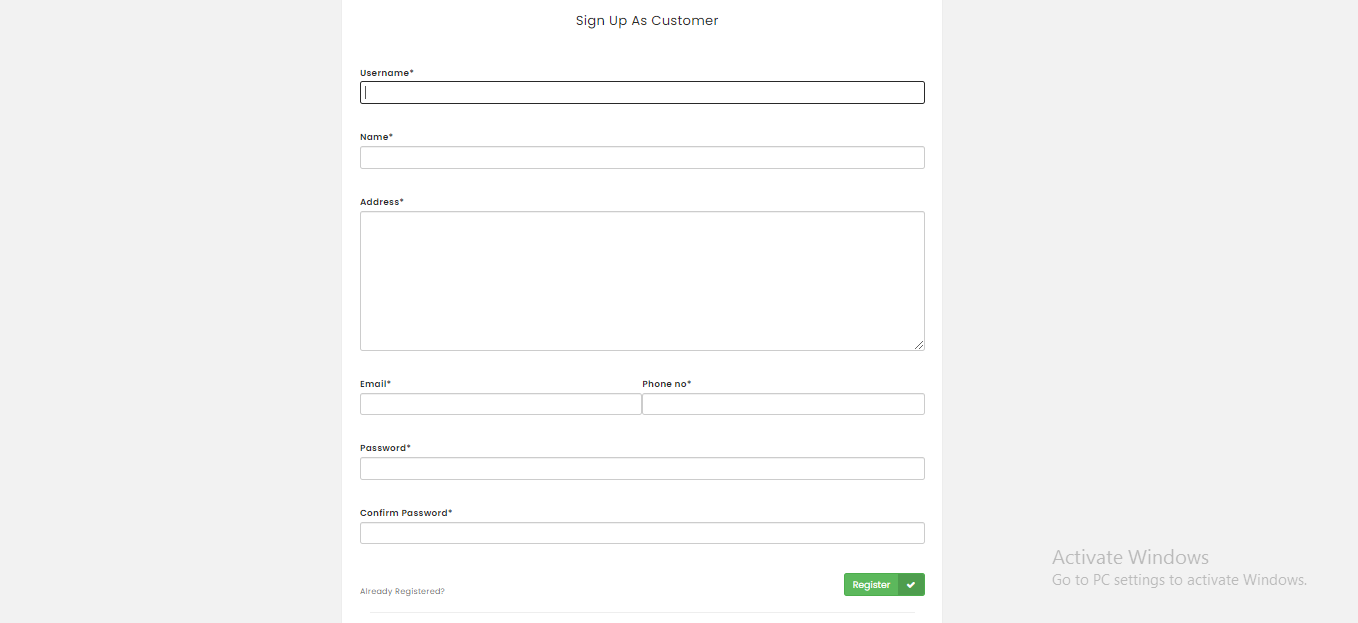
The project is developed using Python, Django and Sqlite as back end. This language selection is based on the requirement specification of the user and analysis of the existing system, with flexibility for future enhancement. Since the system is developed in modules, future enhancement is very easy.

# *SCREENSHOTS*

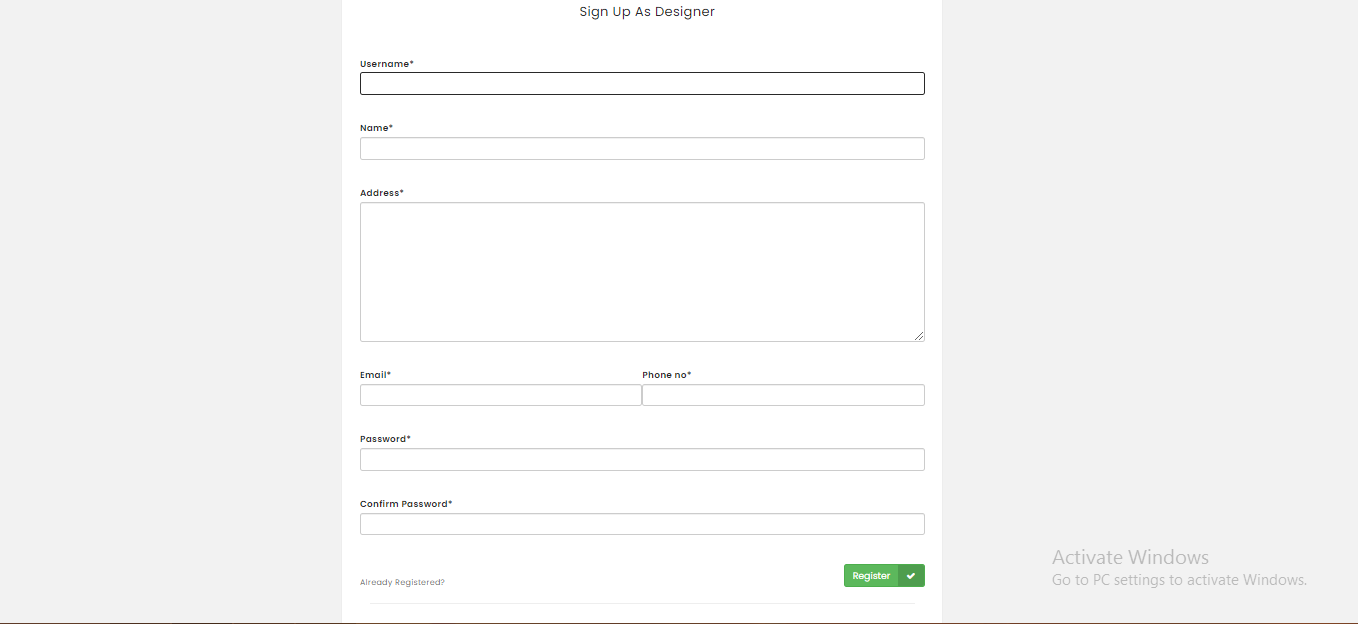
Login page



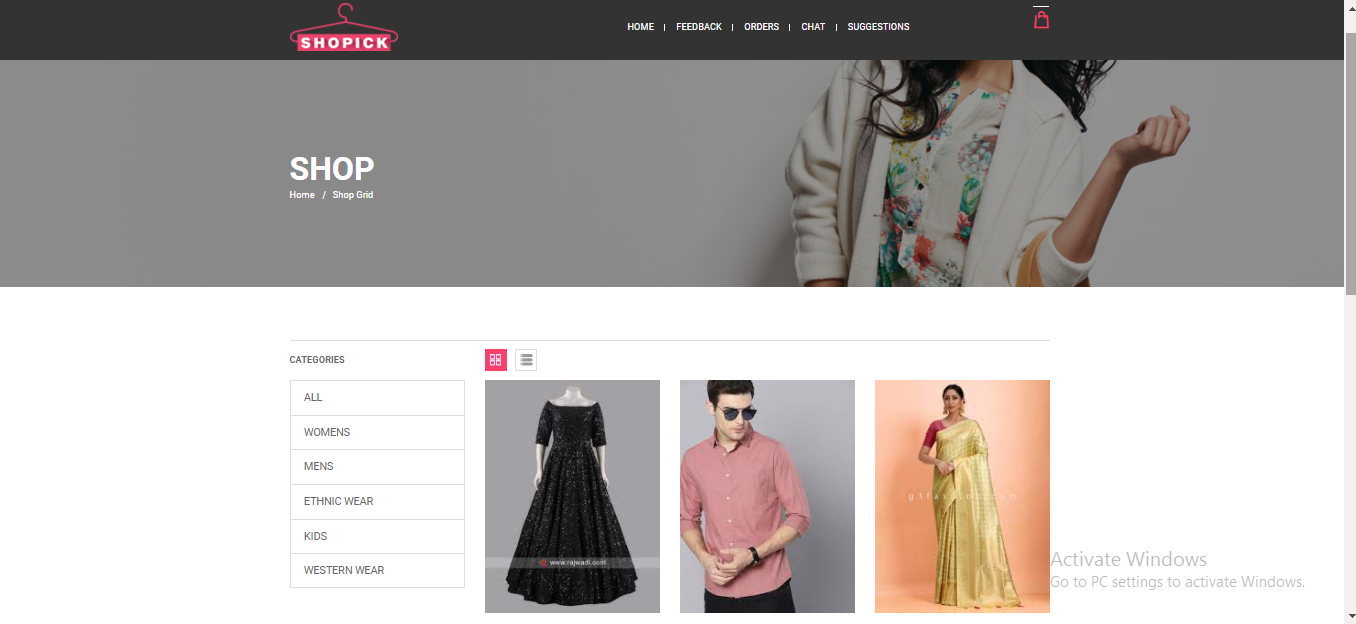
Customer Registration



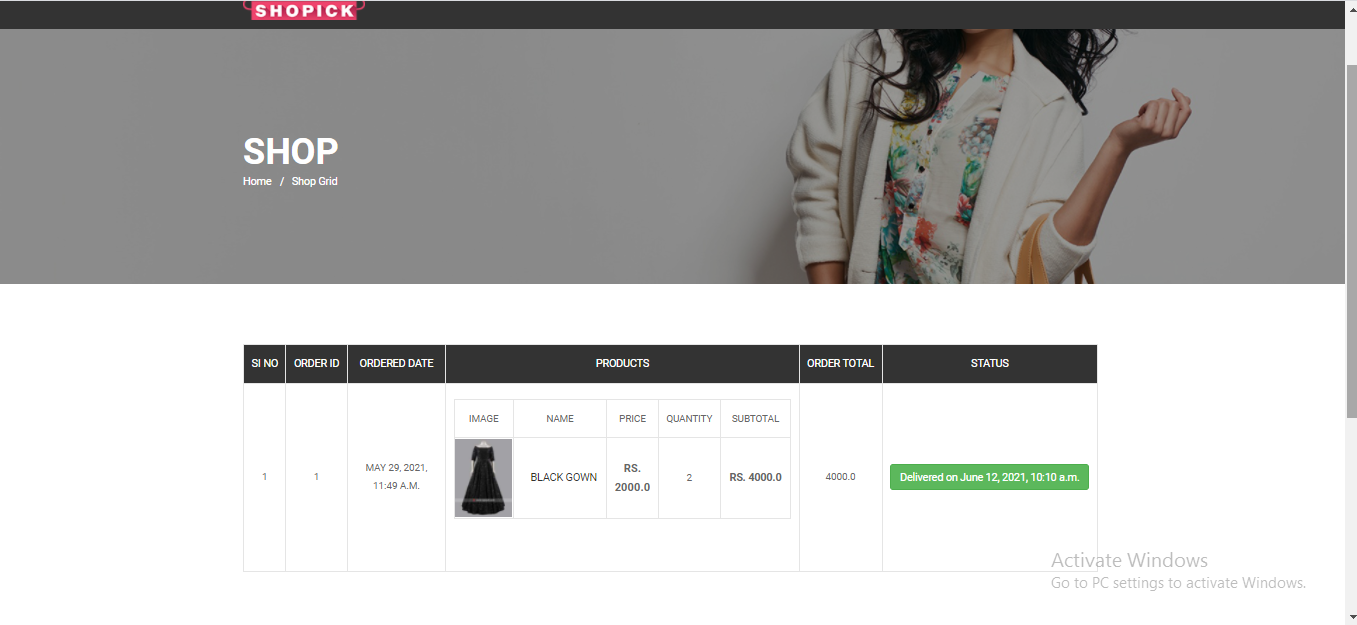
Designer Registration



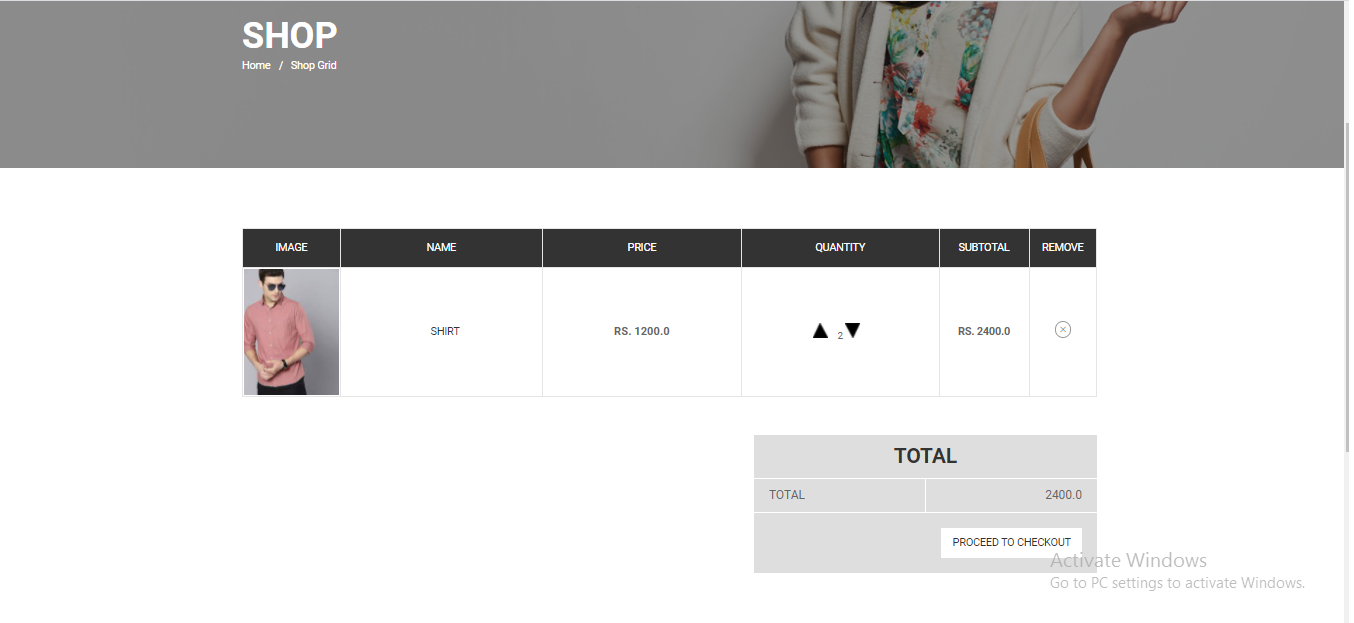
Customer Home



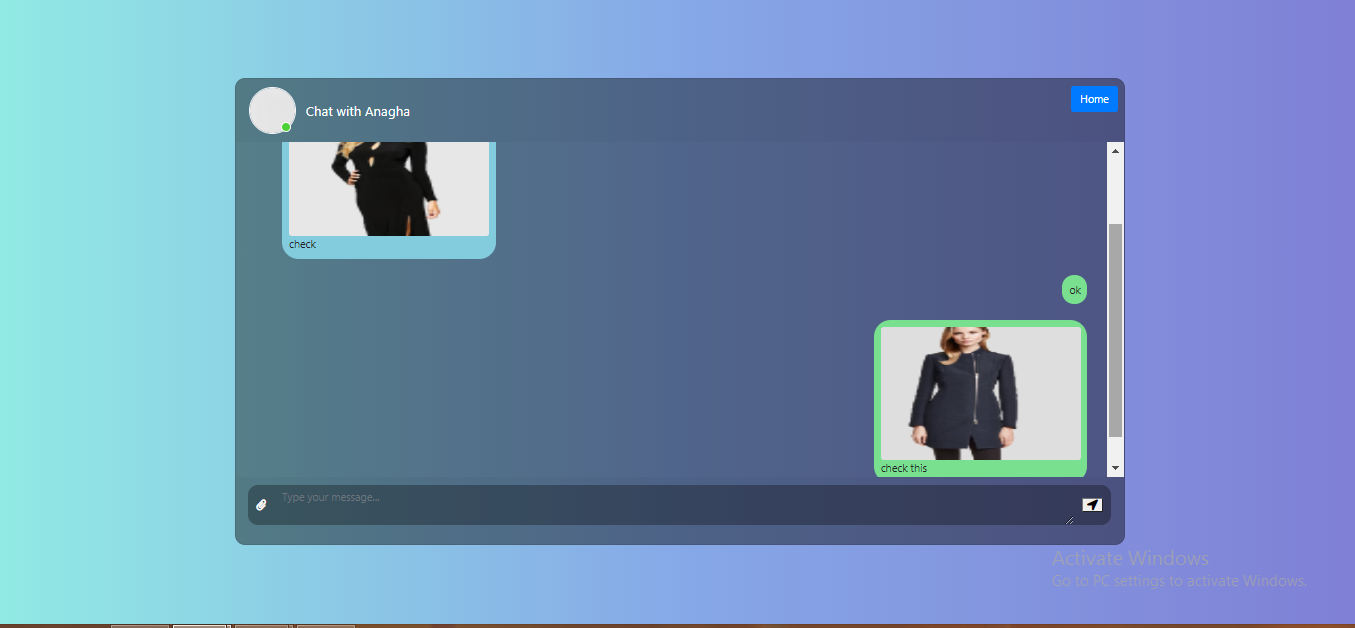
Orders Page



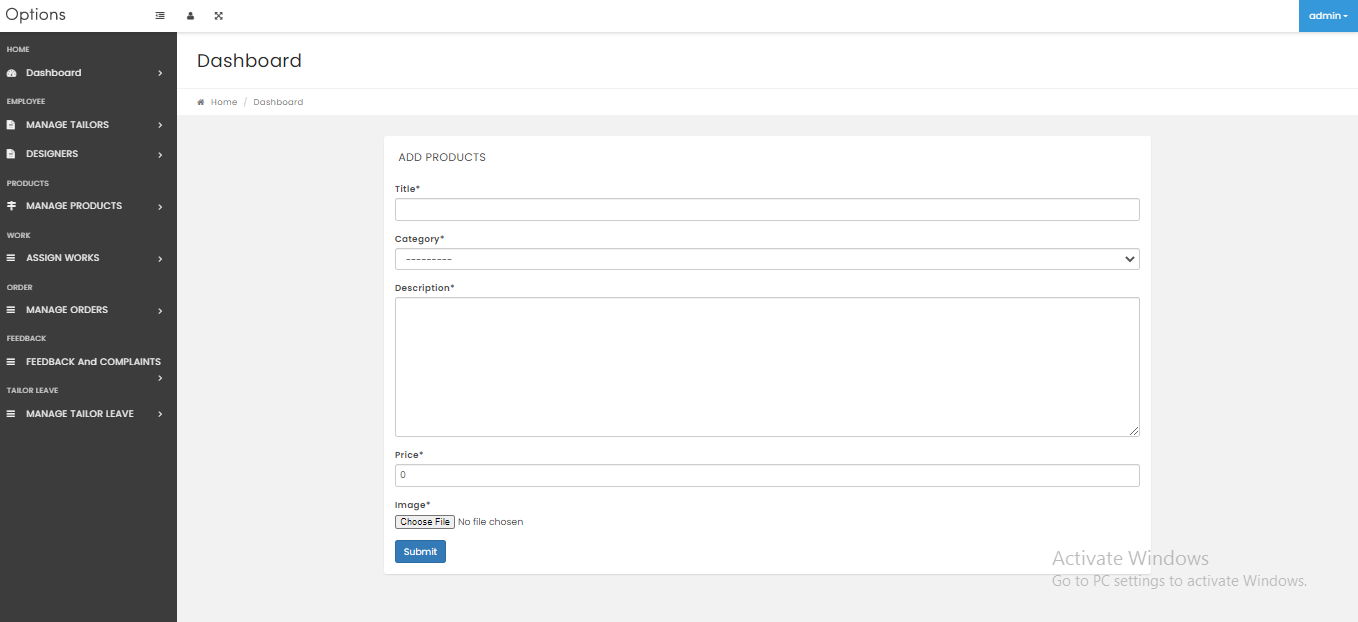
Cart Page



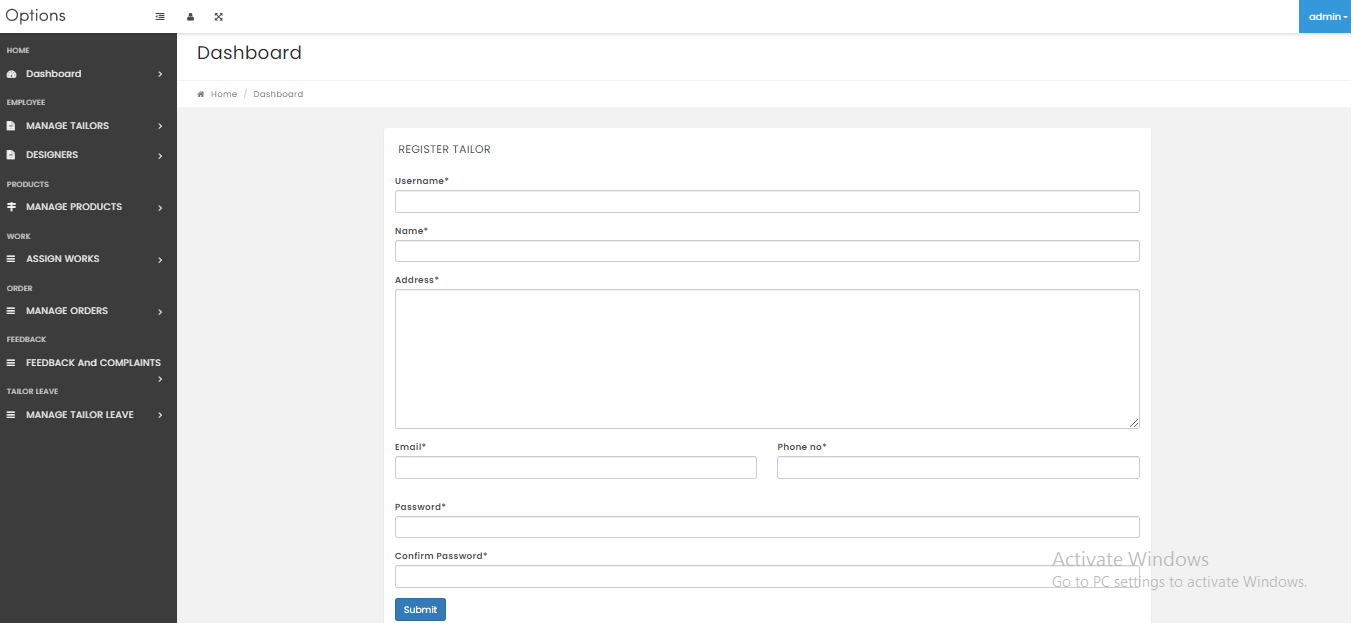
Chatting Page



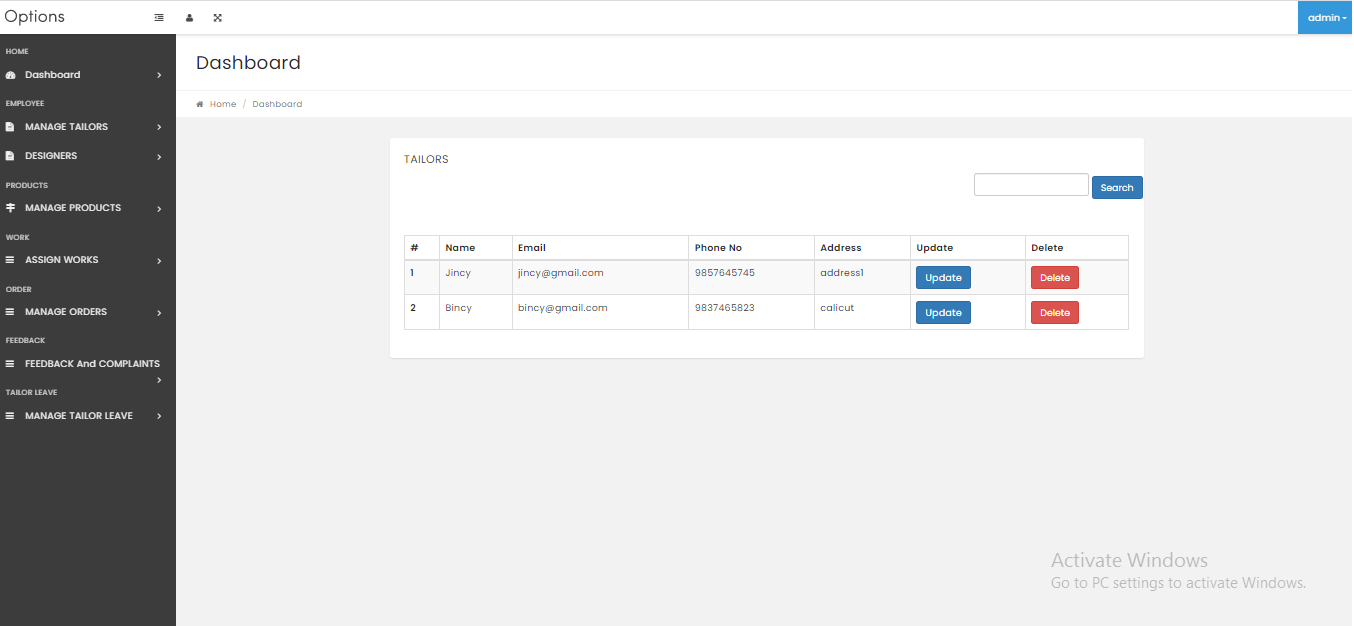
Add Products



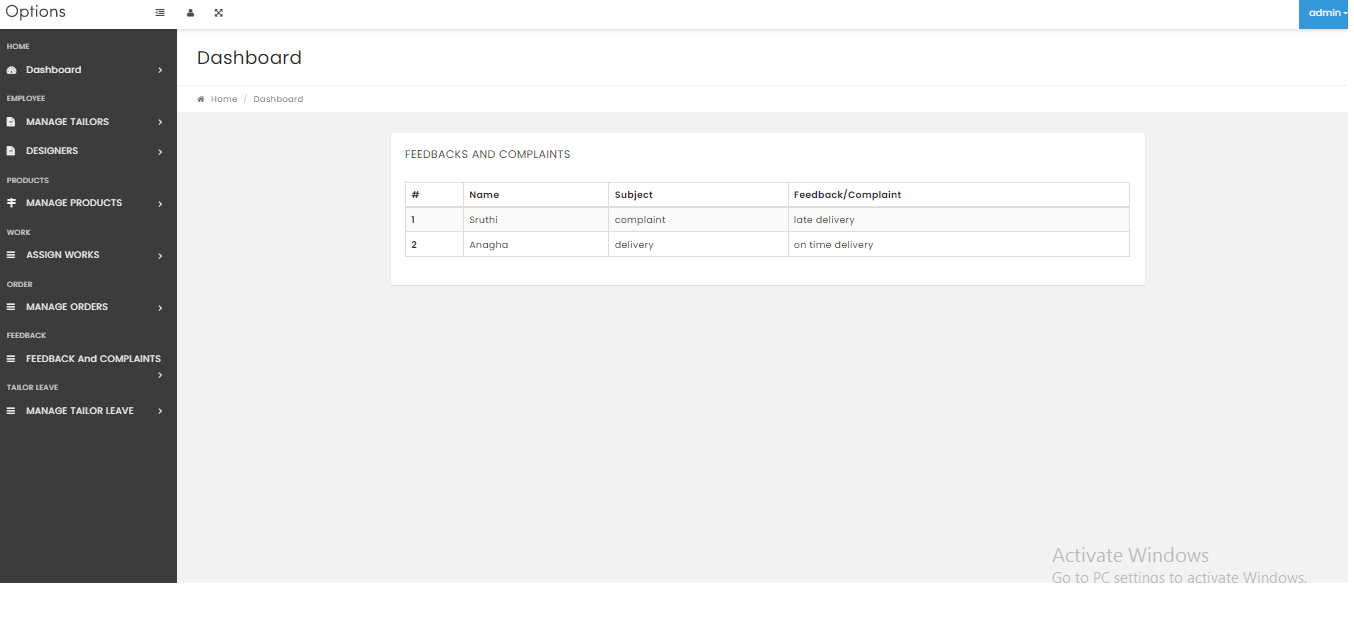
Register Tailor



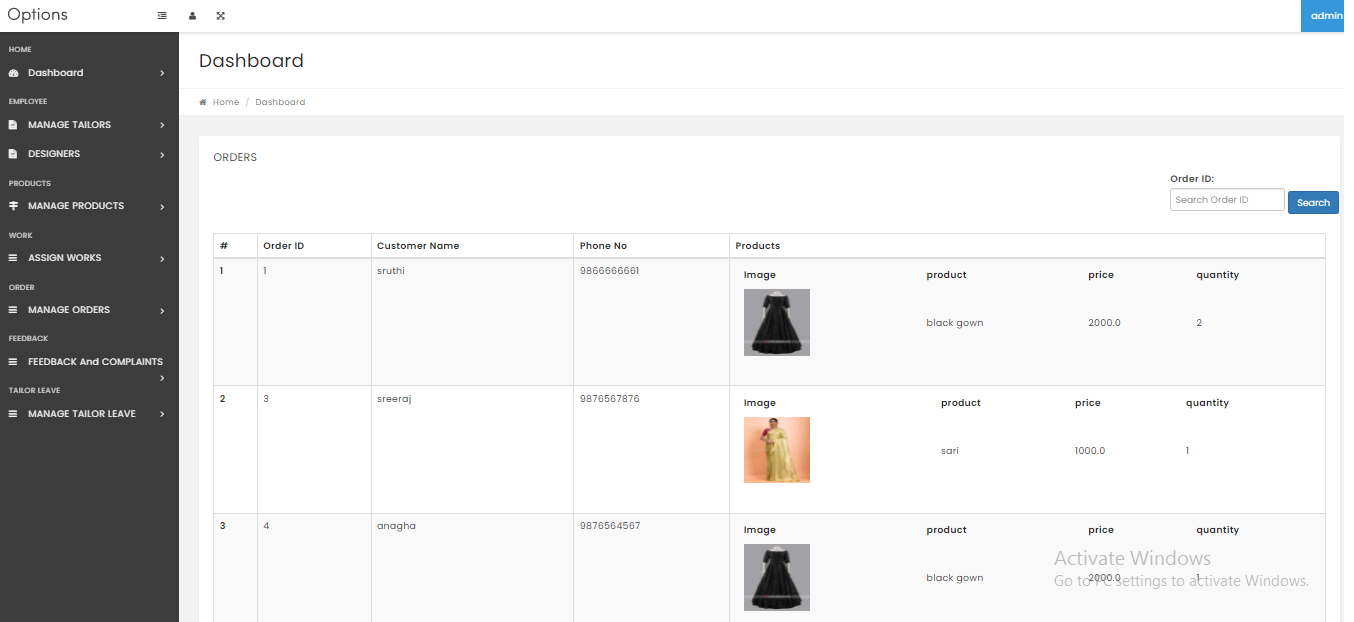
Admin View Tailors



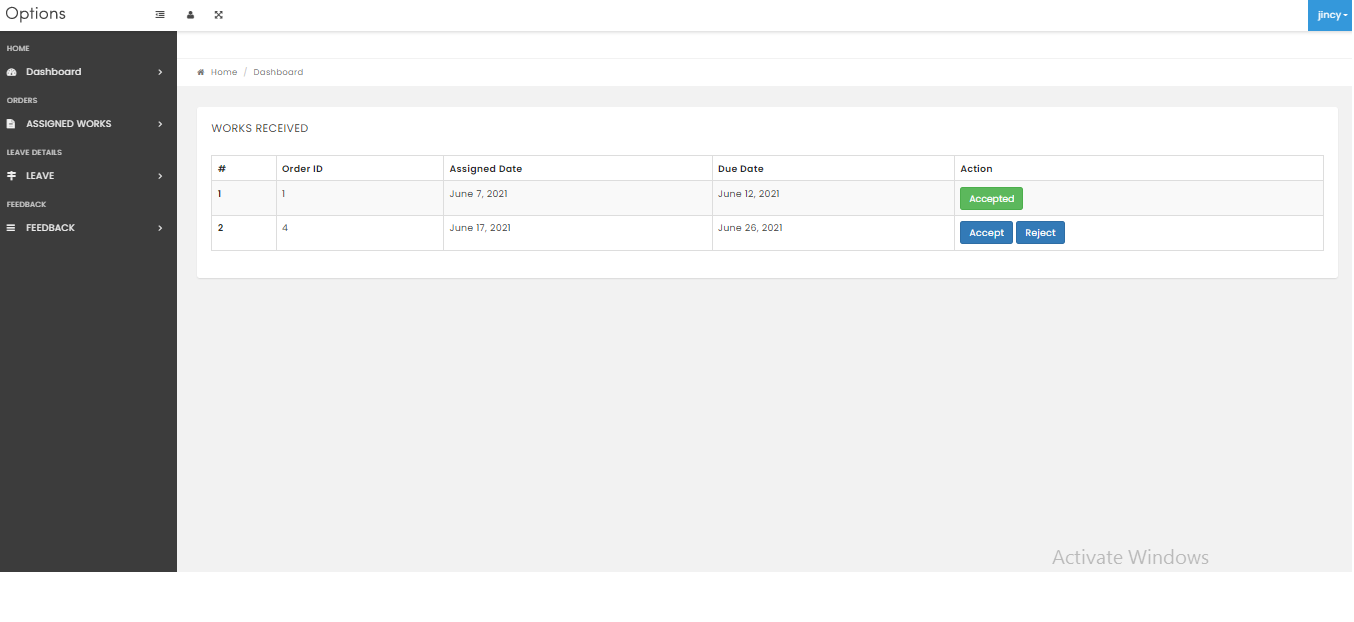
Admin View Feedback



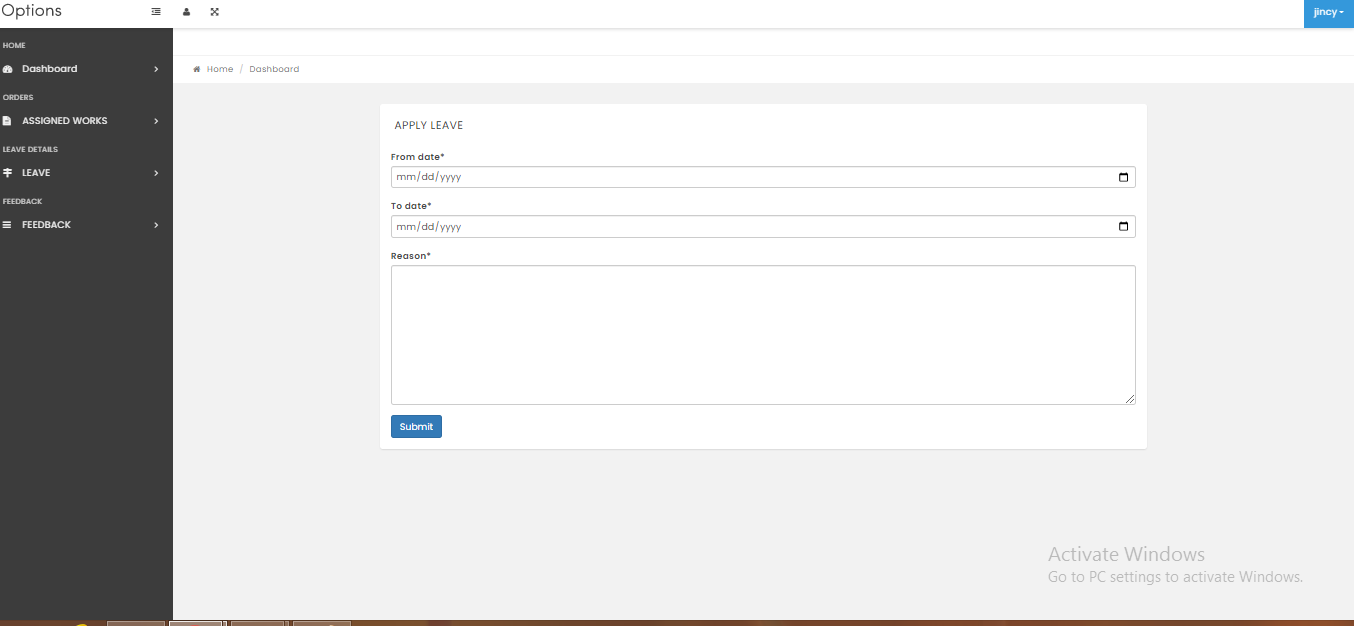
Admin View Products



Tailor view assigned works



Tailor Apply leave



Designer Add Suggestion

