**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

"Jnana Sangama", Belagavi-590 018



A Mini -Project Work on

**“Tailoring Shop Management System”**

A Dissertation work submitted in partial fulfillment of the requirement

for the award of the degree

**Bachelor of Engineering**

In

**Information Science & Engineering**

Submitted by

**Prajwal K V 1AY18IS081**

**Revanth Gowda V K 1AY18IS095**

Under the guidance of

**Prof. Hemantha T D**

Assistant Professor



**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

**ACHARYA INSTITUTE OF TECHNOLOGY**

**(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI )**

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

**2020-21**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

ACHARYA INSTITUTE OF TECHNOLOGY

**(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI)**

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107



**Certificate**

This is to Certify that the Mini-Project work entitled **“Tailoring Shop Management System”** is a bonafide work carried out by **Prajwal K V(1AY18IS081) and Revanth Gowda V K(1AY18IS095)**  in partial fulfillment for the award of the degree of **Bachelor of Engineering** in **Information Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.

**Prof. Hemantha T D Prof. Marigowda C K**

Guide HOD

**Name of the Examiners Signature with date**



**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of this mini-project would be incomplete without the mention of the people who made it possible through constant guidance and encouragement.

We would take this opportunity to express our heart-felt gratitude to **Sri. B. Premnath Reddy**, Chairman, Acharya Institutes and **Dr. Prakash M R**, Principal, Acharya Institute of Technology for providing the necessary infrastructure to complete this mini-project.

We wish to express our deepest gratitude and thanks to **Prof. Marigowda C K**, Head of the Department, Information Science and Engineering.

We wish to express sincere thanks to my guide **Prof. Hemantha T D**, Assistant Professor, Department of Information Science and Engineering for helping us throughout and guiding us from time to time.

A warm thanks to all the faculty of Department of Information Science and Engineering, who have helped us with their views and encouraging ideas.

**ABSTRACT**

The project “**TAILORING SHOP MANAGEMENT SYSTEM**” is aimed to automate the tailoring sector which is manually maintained. After the automation this will mean better services and good keeping of records, data integrity, data security, quick search and also paperless environment. The project has mainly tackled management of information for the customers.

Every user of the system will have to log into the system using username and password so that security and authentication will be ensured. Once logged in, a customer can check dress status or even give feedback. The system administrator is able to manage customer information and also update records.

This will help in realizing the vision 2030 where the government wants its people to be digitally informed and also automate all the government sectors and ministries, hence embracing Electronic Governing.

**TABLE OF CONTENTS**

**Acknowledgement i**

**Abstract ii  
1. Introduction 1**1.1 Introduction to DBMS 1  
 1.1.1 Why DBMS? 2  
 1.1.2 Database applications 2  
 1.1.3 Advantages of DBMS 3  
 1.1.4 Components of DBMS 4  
 1.1.5 Three-Schema architecture 5

**2. System Requirements 6**2.1 Hardware Requirements 6  
2.2 Software Requirements 6

**3. Design 7**3.1 ER Diagram 7  
3.2 Schema Diagram 9

**4. Implementation 10**4.1 Tables 10  
 4.1.1 shop 10  
 4.1.2 customer 11   
 4.1.3 pant 11  
 4.1.4 shirt 12  
 4.1.5 c\_order 13  
 4.1.6 bill 14

4.1.7 logs 15  
4.2 Triggers 15  
4.3 Stored Procedure 16

**5. Snapshots 16  
Conclusion & Future Enhancements 22**  
**Bibliography 23**

**TABLE OF FIGURES**

1.1 Components of Database Management System 4

1.2 Architecture of database system 5

3.1 Entity Relationship Diagram 7

3.2 Schema Diagram 9

5.1 Snapshot of Home page 16

5. 2 Snapshot of customer login page 17

5.3 Snapshot of Tailor login page 17

5.4 Snapshot of admin page 18

5.5 Snapshot of add customer page 18

5.6 Snapshot of new order page 19

5.7 Snapshot of add measurement page 19

5.8 Snapshot of order details page 20

5.9 Snapshot of get c\_id page 20

5.10 Snapshot of customer page 21

**CHAPTER 1**

**INTRODUCTION**

We are in a world where it is not easy for corporations and small enterprises to survive without using computer. Companies that want to grow have to use computerized systems in order to perform their tasks. Saving time and money is a very crucial matter for enterprises therefore companies will make use of computers and internet in order to speed up their transactions. In the era we are living now, it is very difficult for an enterprise that does not use technology means to compete with other enterprises. Not using computerized system will let you behind others and will slow down your enterprise. Paper-based systems have no place nowadays. Recording data on paper is very obsolete. A very efficient to keep data is to create an online database. Once the data are saved in a database, we can access the data anywhere anytime. The information retrieved from the database with easiness and within seconds. As a company there it is important to have a database in order to keep data regarding your customers. Business depends on customers therefore we need to use technology tools to collect data of the clients.

Enterprises want to complete their work faster, so interactive, responsive and efficient applications are decisive. People love to use systems that are user-friendly with a nice interface.

Looking at the impact of technology on business, it is very important for small and big businesses to use computerized systems.

Like others entrepreneurs, it’s efficient and interesting for the tailors to use computerized system to perform their daily tasks. A system that is interactive, fast that allows them to do tasks quickly and with easiness.

* 1. **Introduction to DBMS**

DBMS stands for **D**ata**b**ase **M**anagement **S**ystem. We can break it like this DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Basically DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

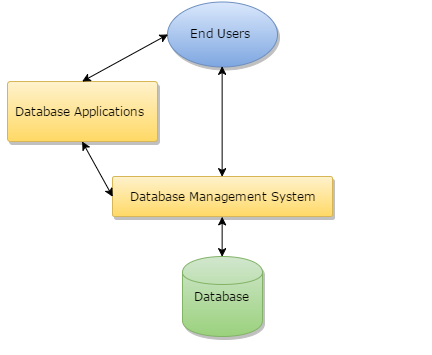
The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of data. Meaningful data combined to form information. Hence, information is interpreted data – data provided with semantics. MS. ACCESS is one of the most common examples of database management software.

Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

* + 1. **Why DBMS?**
* To develop software applications in less time.
* Data Independence and efficient use of data.
* For uniform data administration.
* For data integrity and security.
* For concurrent access of data, and data recovery from crashes.
* To use user-friendly declarative query language.
  + 1. **Database applications**
* **Telecom:** There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
* **Industry:** Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.
* **Education sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.
* **Online shopping:** You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.
* **Banking system:** For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
  + 1. **Advantages of DBMS**

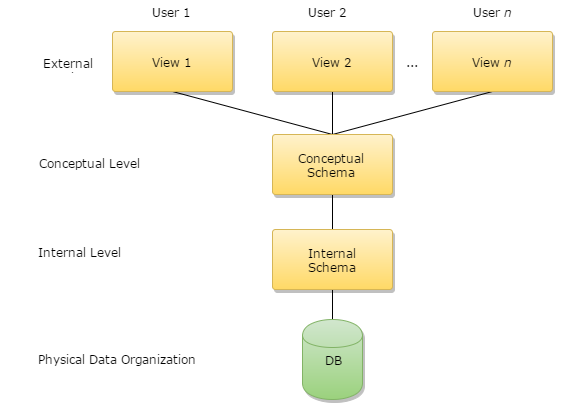
A DBMS manage data and has many advantages.

* **Data Independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
* **Efficient data access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
* **Data integrity and security:**  If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.
* **Data administration:** When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.
* **Providing backup and recovery:** A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery.
* **Permitting inferencing and actions using rules:** Some database systems provide capabilities for defining deduction rules for inferencing new information from the stored database facts.
  + 1. **Components of DBMS**

****

**Fig-1.1: Components of a Database Management System**

* **Users:** Users may be of any kind such as DB administrator, System developer or database users.
* **Database application:** Database application may be Departmental, Personal, organization’s and / or Internal.
* **DBMS:** Software that allow users to create and manipulate database access.
* **Database:** Collection of logical data as a single unit.
* **Database access language:** This is used to access the data to and from the database, to enter new data, update existing data, or retrieve required data from databases. The user writes a set of appropriate commands in a database access language, submits these to the DBMS, which then processes the data and generates and displays a set of results into a user readable form.
  + 1. **Three-Schema architecture**

****

**Fig-1.2: Architecture of database system**

The levels form a three-level architecture that includes an external, a conceptual, and an internal level. The way users recognize the data is called the external level. The way the DBMS and the operating system distinguish the data is the internal level, where the data is actually stored using the data structures and file. The conceptual level offers both the mapping and the desired independence between the external and internal levels.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

**2.1 Hardware Requirements**

* **Processor:** Intel Core2 Quad @ 2.4Ghz on Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
* **RAM:** 2GB of RAM
* **Memory:** 256GB Hard drive
* **Keyboard:** MS compatible keyboard
* **Mouse:** MS compatible mouse

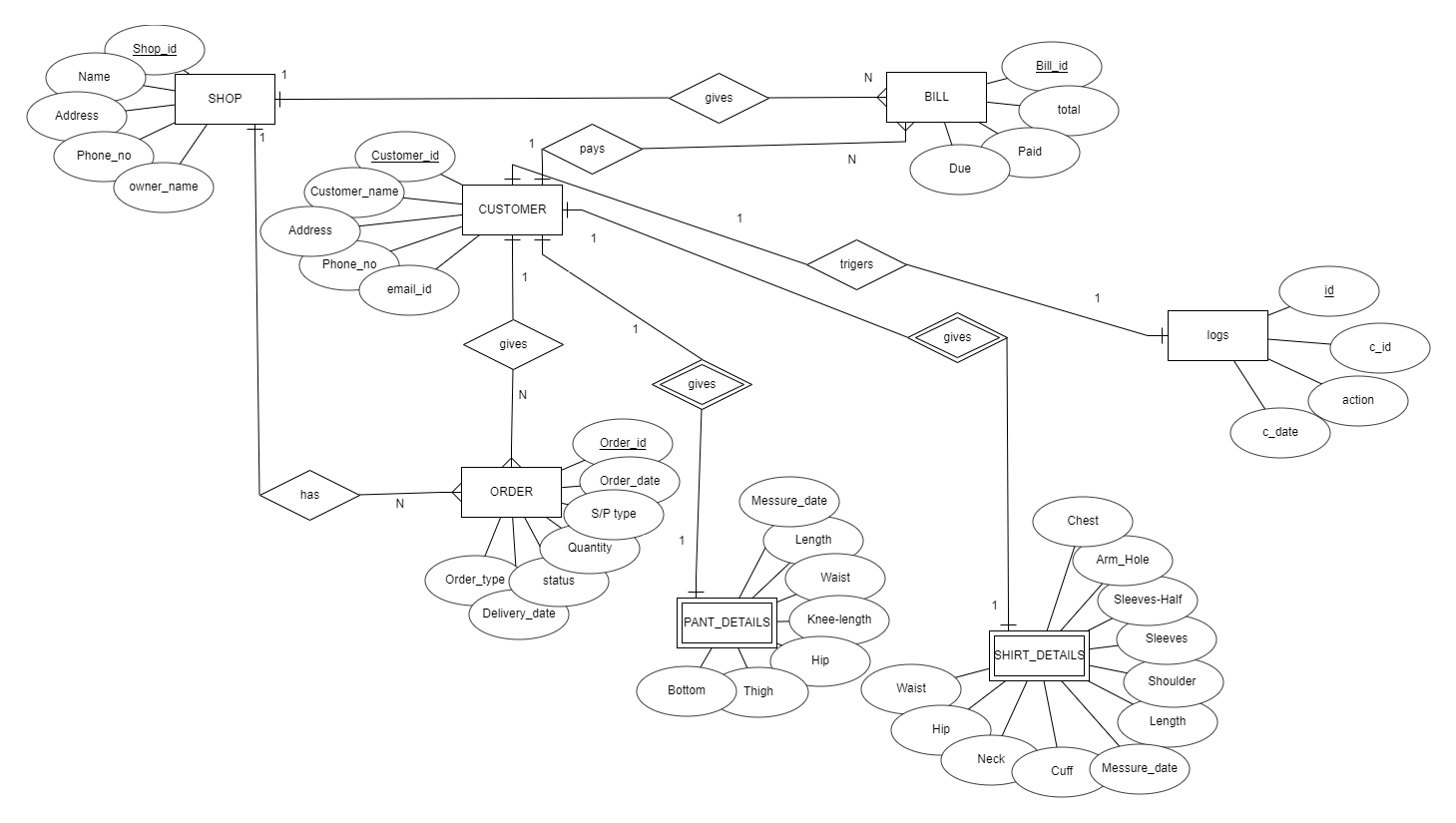
**2.2 Software Requirements**

* **Operating system:** Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit / Windows® 10 32/64-Bit.
* **Front end:** Html, CSS, Bootstrap, JavaScript
* **Back end:** MySQL Database
* **Server Side Scripting:** PHP
* **Web Server:** XAMPP OR MAMP

**CHAPTER 3**

**DESIGN**

**3.1 ER Diagram**

****

**Fig-3.1: Entity Relationship Diagram**

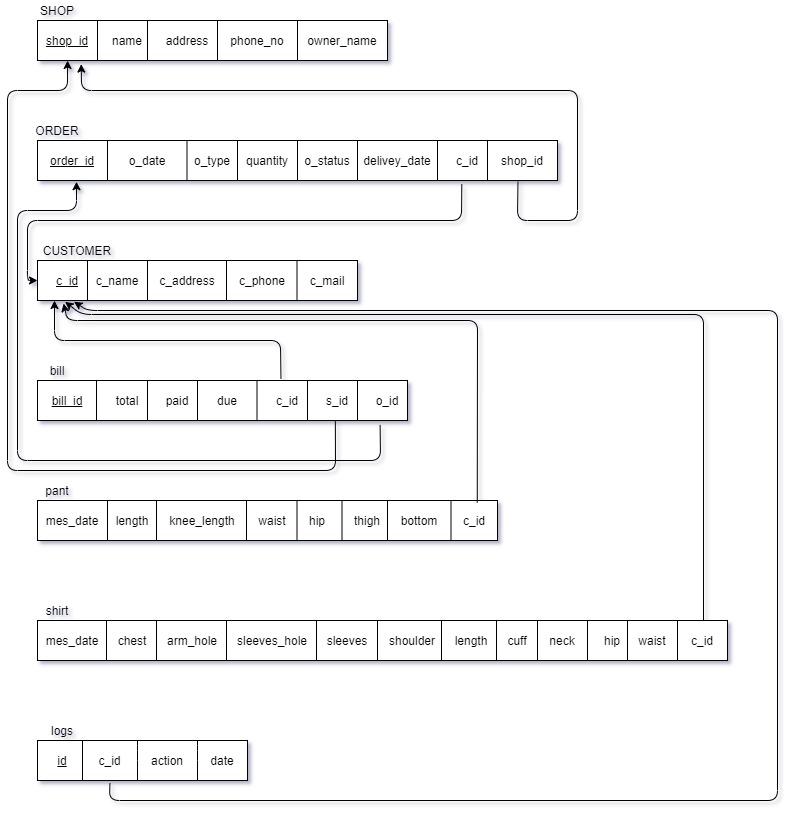
**1:N**

* One shop has many orders.
* One customer gives many orders.
* One customer pays many bills.
* One shop gives many bills.

**1:1**

* One customer has one shirt details.
* One customer has one pant details.
* One customer triggers one logs.

**3.2 Schema Diagram**

****

**Fig-3.2: Schema Diagram**

**Schema Diagram:** An illustrative display of (most aspects of) a database schema.

**Schema Construct:** A component of the schema or an object within the schema,

e.g., STUDENT, COURSE.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 Tables**

**4.1.1 shop**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | s\_id | Integer | Primary Key |
| 2 | s\_name | Varchar |  |
| 3 | s\_address | Varchar |  |
| 4 | s\_phone | Bigint |  |
| 5 | s\_owner | Varchar |  |
| 6 | s\_password | Varchar |  |

CREATE TABLE shop (

s\_id INTEGER AUTO\_INCREMENT PRIMARY KEY,

s\_name VARCHAR (40),

s\_address VARCHAR (50),

s\_phone BIGINT,

s\_owner VARCHAR (20),

s\_password VARCHAR (100)

);

**4.1.2 customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | c\_id | Integer | Primary Key |
| 2 | c\_name | Varchar |  |
| 3 | c\_address | Varchar |  |
| 4 | c\_phone | Bigint |  |
| 5 | c\_mail | Varchar |  |

CREATE TABLE customer (

c\_id INTEGER AUTO\_INCREMENT PRIMARY KEY,

c\_name VARCHAR (20),

c\_address VARCHAR (50),

c\_phone BIGINT UNIQUE,

c\_mail VARCHAR (30),

);

**4.1.3 pant**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | mes\_date | Date |  |
| 2 | length | Integer |  |
| 3 | waist | Integer |  |
| 4 | knee\_length | Integer |  |
| 5 | hip | Integer |  |
| 6 | thigh | Integer |  |
| 7 | bottom | Integer |  |
| 8 | c\_id | Integer | Foreign Key to customer |

CREATE TABLE pant (

mes\_date DATE,

length INTEGER,

waist INTEGER,

knee\_length INTEGER,

hip INTEGER,

thigh INTEGER,

bottom INTEGER,

c\_id INTEGER,

CONSTRAINT FOREIGN KEY (c\_id) REFERENCES customer(c\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

**4.1.4 shirt**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | mes\_date | Date |  |
| 2 | chest | Integer |  |
| 3 | arm\_hole | Integer |  |
| 4 | sleeves\_half | Integer |  |
| 5 | sleeves | Integer |  |
| 6 | shoulder | Integer |  |
| 7 | length | Integer |  |
| 8 | cuff | Integer |  |
| 9 | neck | Integer |  |
| 10 | hip | Integer |  |
| 11 | waist | Integer |  |
| 12 | c\_id | Integer | Foreign Key to customer |

CREATE TABLE shirt (

mes\_date DATE,

chest INTEGER,

arm\_hole INTEGER,

sleeves\_half INTEGER,

sleeves INTEGER,

shoulder INTEGER,

length INTEGER,

cuff INTEGER,

neck INTEGER,

hip INTEGER,

waist INTEGER,

c\_id INTEGER,

CONSTRAINT FOREIGN KEY (c\_id) REFERENCES customer(c\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

**4.1.5 c\_order**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | o\_id | Integer | Primary Key |
| 2 | o\_date | Date |  |
| 3 | o\_type | Varchar |  |
| 4 | quantity | Integer |  |
| 5 | o\_status | Varchar |  |
| 6 | delivery\_date | Date |  |
| 7 | c\_id | Integer | Foreign Key to customer |
| 8 | s\_id | Integer | Foreign Key to shop |

CREATE TABLE c\_order (

o\_id INTEGER AUTO\_INCREMENT,

PRIMARY KEY (o\_id),

o\_date DATE,

o\_type VARCHAR (20),

quantity INTEGER,

o\_status VARCHAR (10),

delivery\_date DATE,

c\_id INTEGER,

s\_id Integer,

CONSTRAINT FOREIGN KEY (c\_id) REFERENCES customer(c\_id)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FOREIGN KEY (s\_id) REFERENCES shop (s\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

**4.1.6 bill**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | bill\_id | Integer | Primary Key |
| 2 | total | Integer |  |
| 3 | paid | Integer |  |
| 4 | due | Integer |  |
| 5 | c\_id | Integer | Foreign Key to customer |
| 6 | s\_id | Integer | Foreign Key to shop |
| 7 | o\_id | Integer | Foreign Key to c\_order |

CREATE TABLE bill(

bill\_id INTEGER AUTO\_INCREMENT,

total INTEGER,

paid INTEGER,

due INTEGER,

c\_id INTEGER,

s\_id INTEGER,

o\_id INTEGER,

PRIMARY KEY (bill\_id),

CONSTRAINT FOREIGN KEY (c\_id) REFERENCES customer(c\_id)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FOREIGN KEY (s\_id) REFERENCES shop (s\_id)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FOREIGN KEY (o\_id) REFERENCES c\_order(o\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

**4.1.7 logs**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **COLUMN\_NAME** | **DATA\_TYPE** | **DESCRIPTION** |
| 1 | id | Integer | Primary Key |
| 2 | c\_id | Integer | Foreign Key to customer |
| 3 | action | Varchar |  |
| 4 | cdate | DATETIME |  |

CREATE TABLE logs(

id INTEGER AUTO\_INCREMENT,

c\_id INTEGER,

action VARCHAR (20),

cdate DATETIME,

PRIMARY KEY (id),

CONSTRAINT FOREIGN KEY (c\_id) REFERENCES customer(c\_id)

ON DELETE CASCADE ON UPDATE CASCADE,

);

**4.2 Triggers**

CREATE TRIGGER `insertLog`

AFTER INSERT ON `customer`

FOR EACH ROW

INSERT INTO logs VALUES (null, NEW. c\_id, "Inserted", NOW())

**4.3 Stored Procedures**

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `getShopId`()

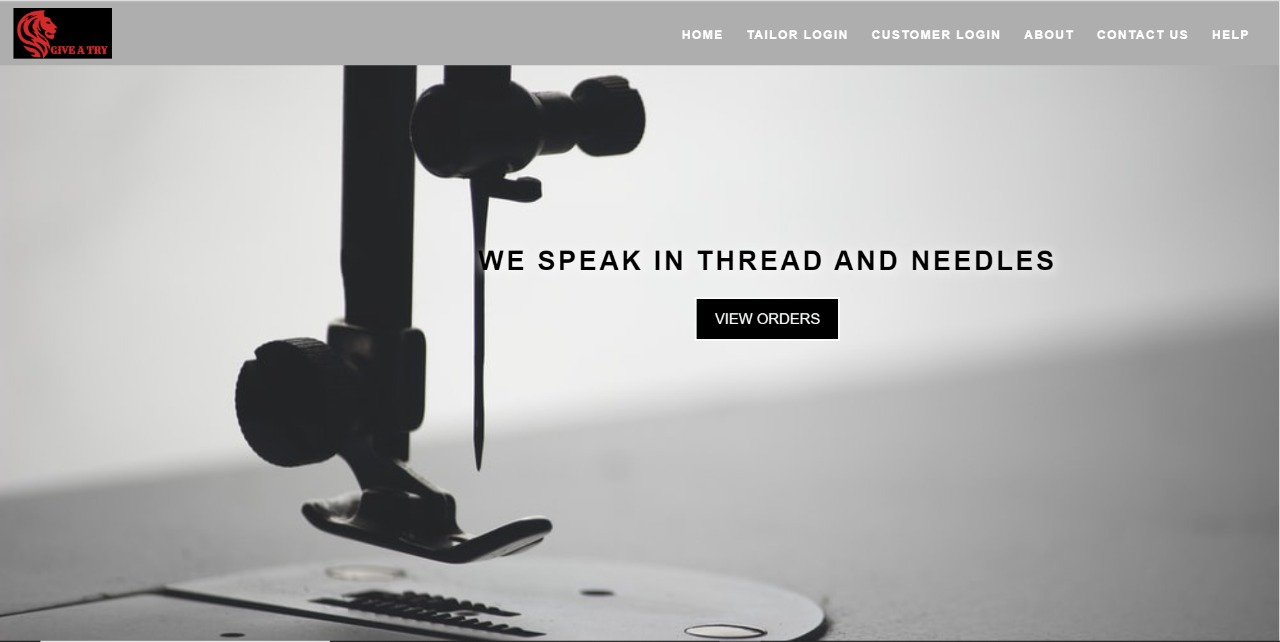
SELECT s\_id FROM `shop`$$

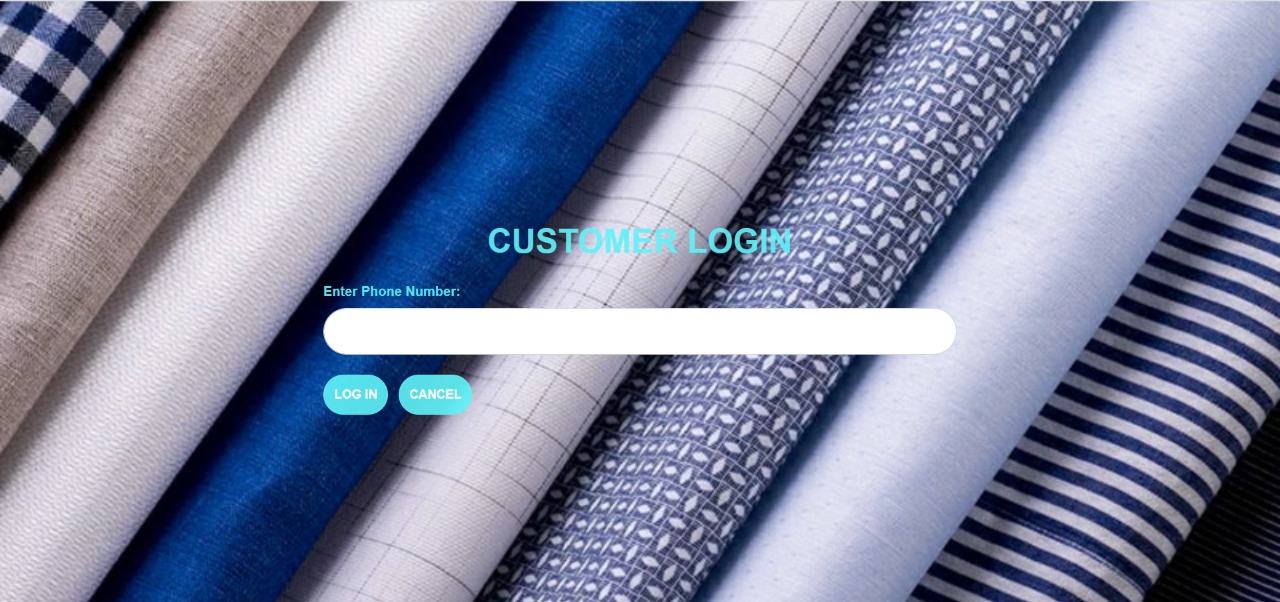
DELIMITER ;

**CHAPTER 5**

**SNAPSHOTS**

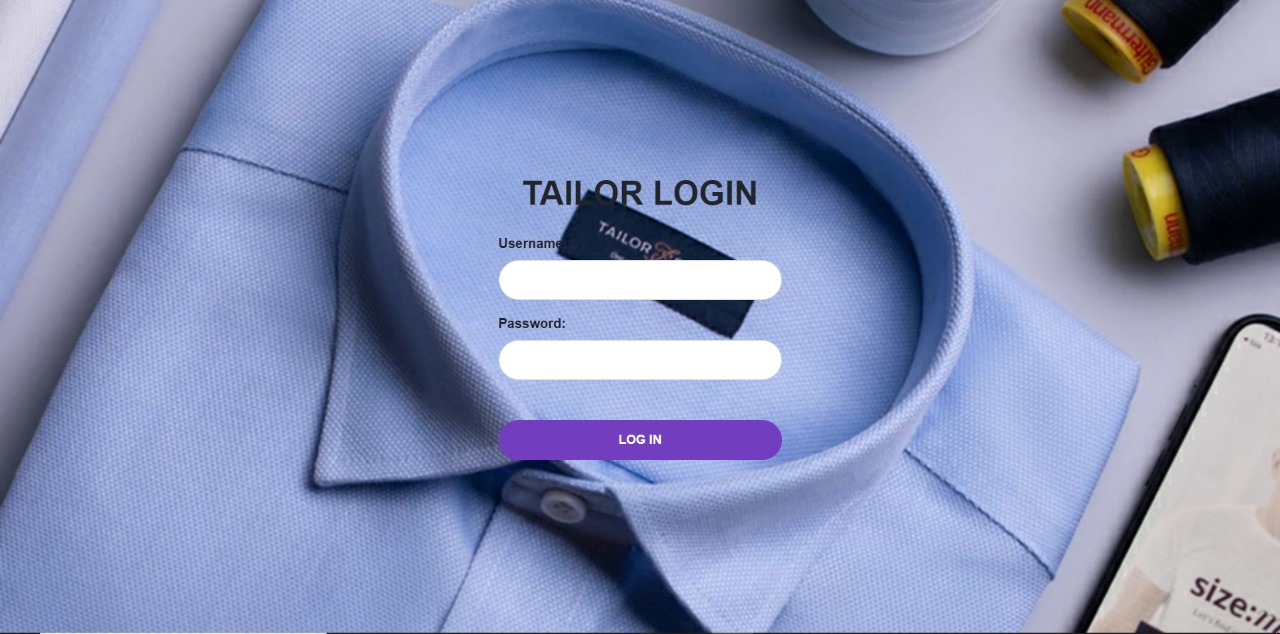
The following snapshot contains the Home page of the application where customer can view details about the Tailor shop**.**

**  
Fig-5.1: Snapshot of Home Page**

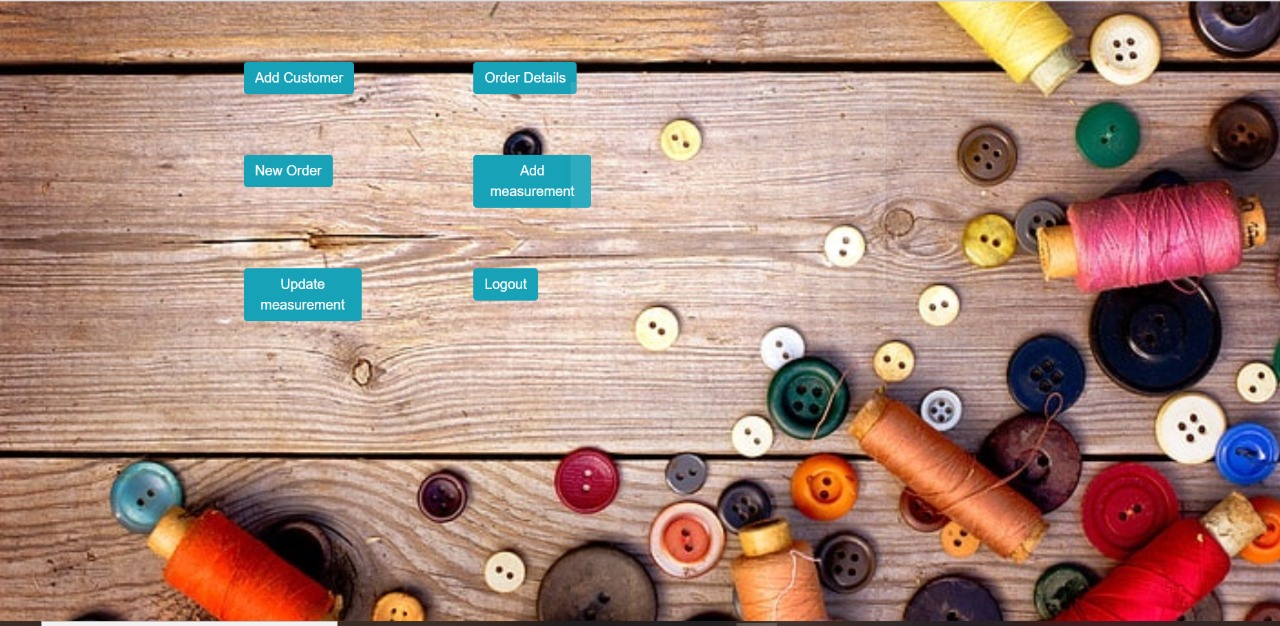
The following snapshot contains the Customer login page of the application. Where the customer uses there phone number to login****

**Fig-5.2: Snapshot of Customer login page**

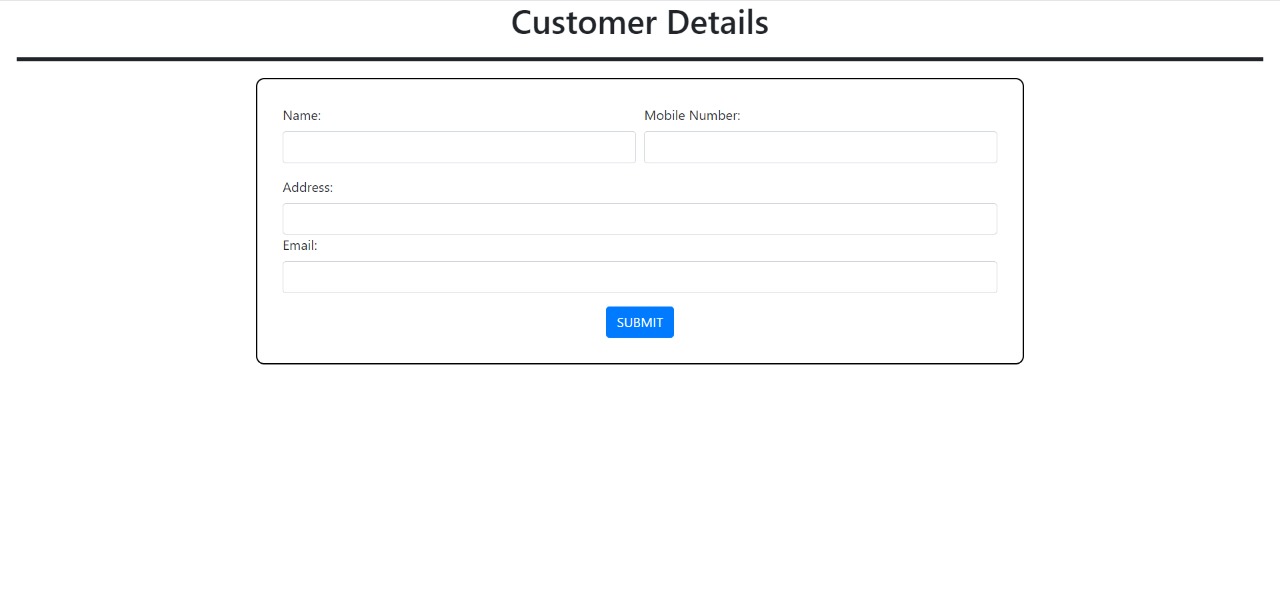
The following snapshot contains the Tailor login page of the application. Where the Tailor enters Username: ”Revanth” and Password: ”kvp@2000” to login.

  
**Fig-5.3: Snapshot of Tailor login page**

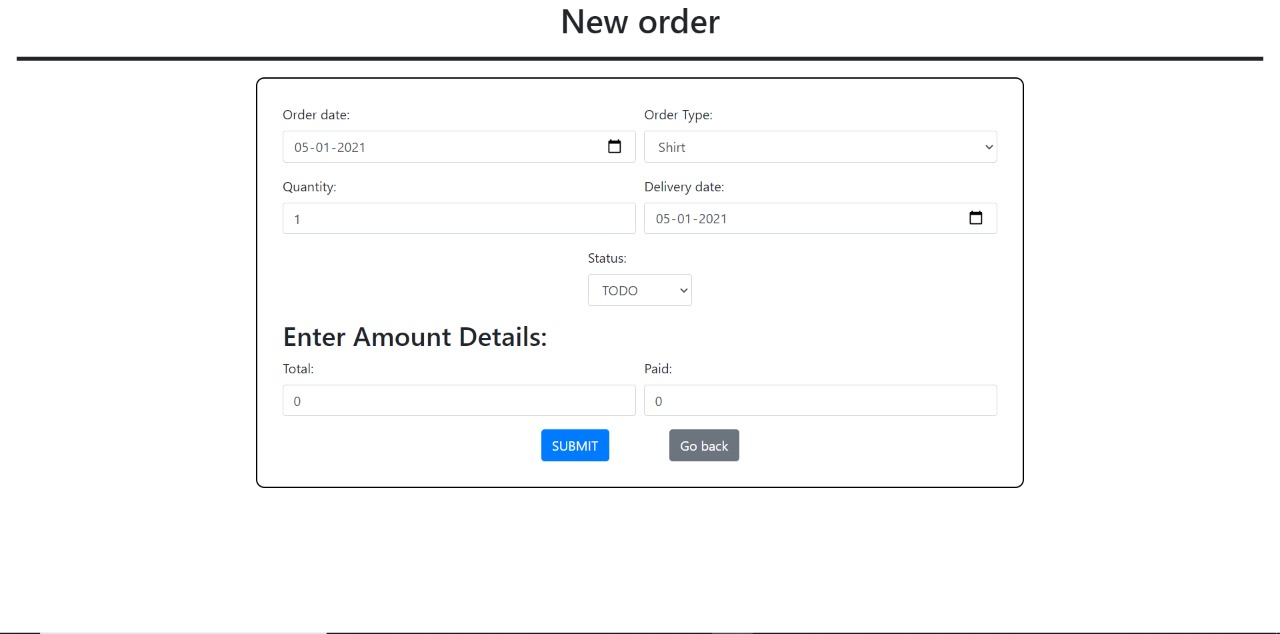
The following snapshot contains the admin page section .Where the Tailor logged in can add a customer, check order details, add a new order, add measurements, update measurements and logout. This is the page that links to all other pages.

**  
Fig-5.4: Snapshot of admin page**

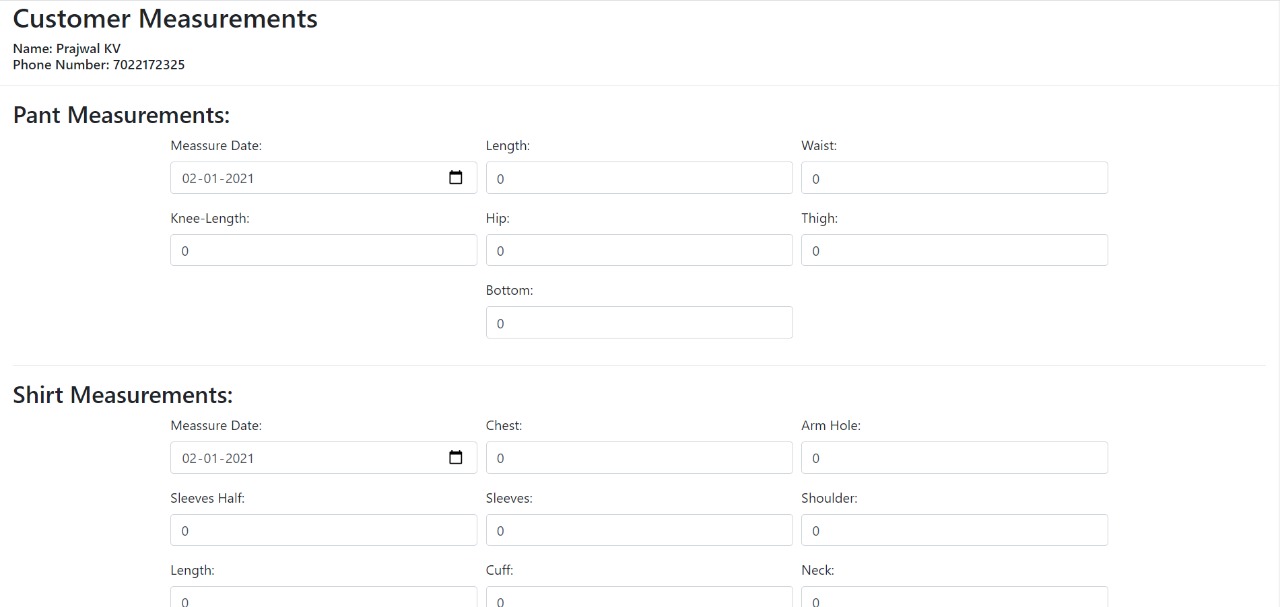
The following snapshot contains add customer page where the details of the customers like name, mobile number, address and email will be added into the database by Tailor.

**  
Fig-5.5: Snapshot of add customer page**

The following snapshot contains new order page where the order details of the customers like order date, order type, quantity, delivery date, status, total amount and paid amount will be added into the database by Tailor**.**

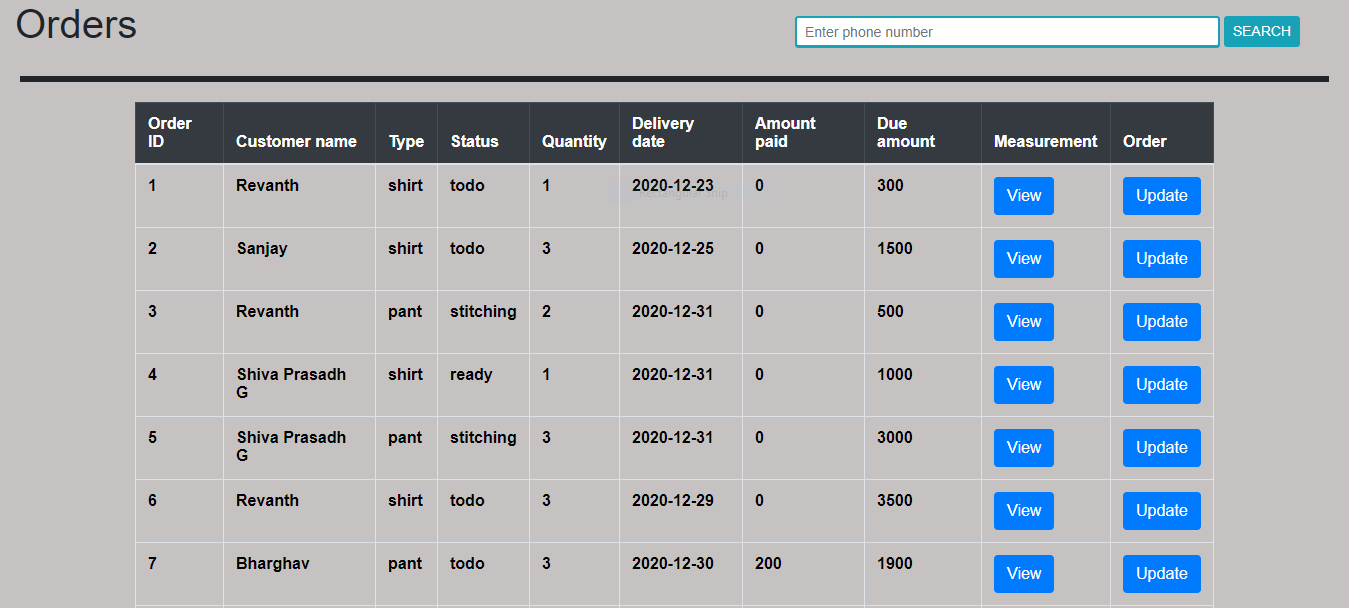
**  
Fig-5.6: Snapshot of new order page**

The following snapshot contains add measurement page where pant and shirt measurement details of the customers with the measurement taken date will be added into the database by Tailor.

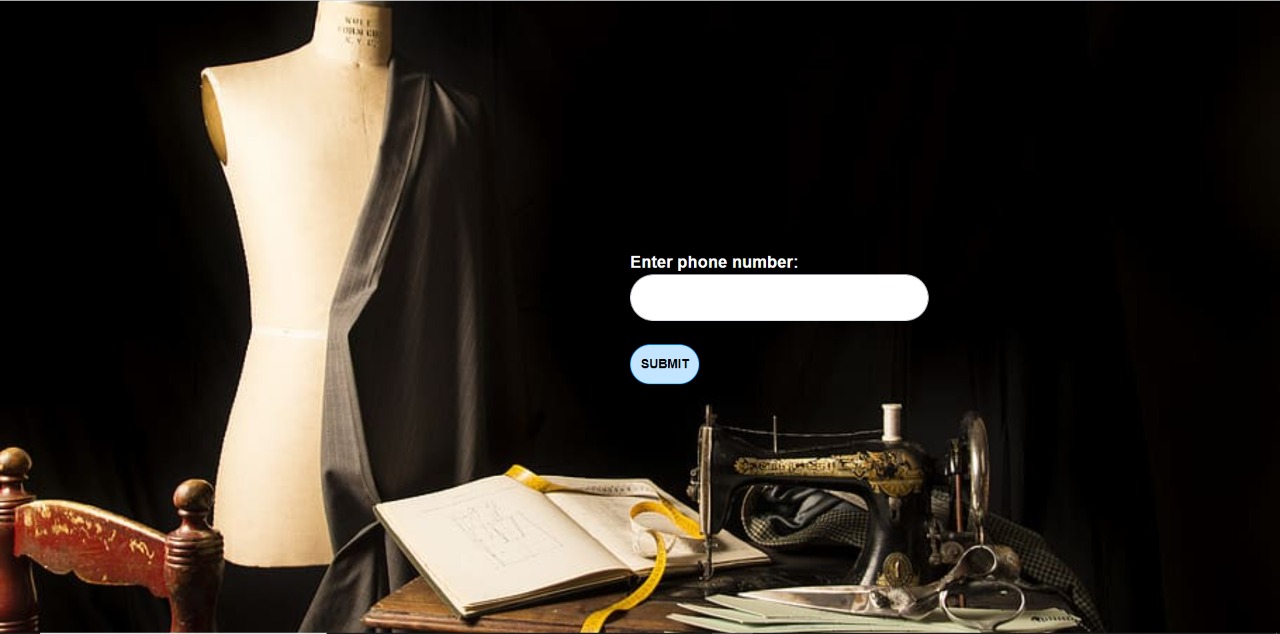
****

**Fig-5.7: Snapshot of the add measurement page**

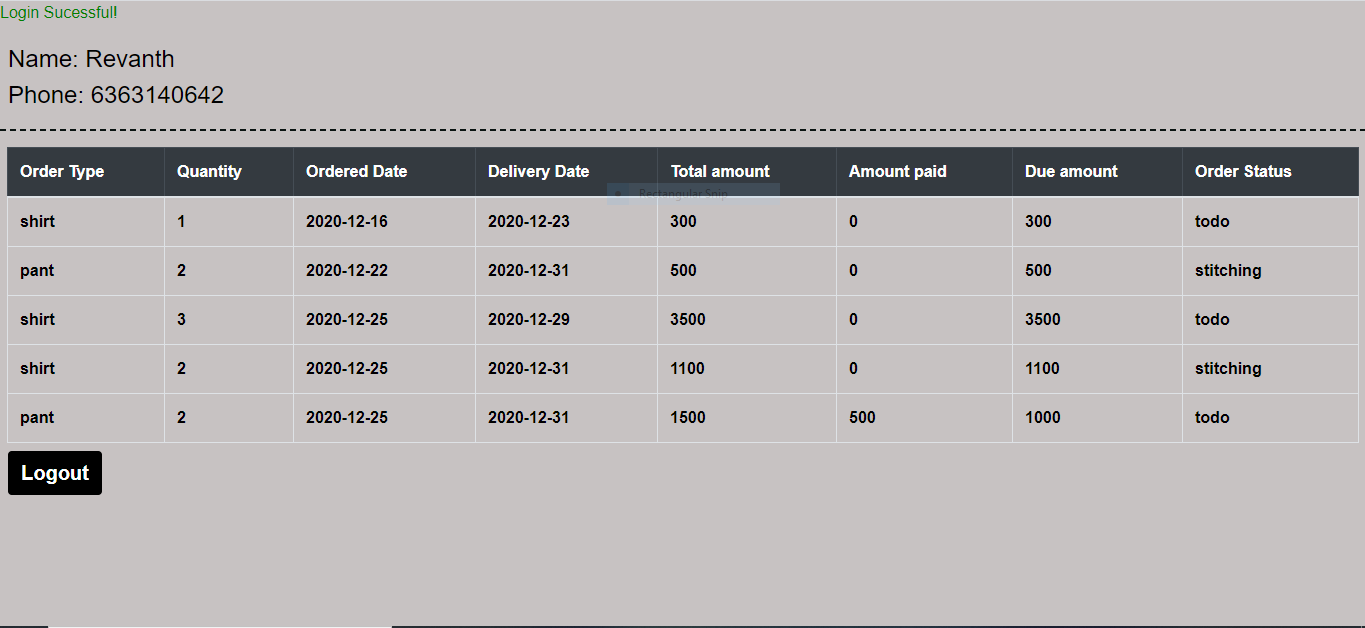
The following snapshot displays the order details of the customers. It also has the option of viewing and updating measurement and order details. Even the amount to be paid and the due amount are displayed.

**  
Fig-5.7: Snapshot of order details page**

The following snapshot contains get\_cid page where the phone number of the customer is entered to retrieve the details of the customer like name, measurements etc. This page helps in altering the customer details by retrieving the data of a particular customer based on the entered phone number.

**   
Fig-5.8: Snapshot of get\_cid page**

The following snapshot displays customer page where the customer can view his order details, amount to be paid, date of delivery and order status.

**  
Fig-5.9: Snapshot of customer page**

**CONCLUSION & FUTURE ENHANCEMENT**

**Conclusion**

The general objective of this project is to develop an interactive and online tailor system that helps tailors to carry out their daily tasks. It is a better system compared to paper-based system. Tailors will create profiles for each customer and record all useful information regarding the clients. All the details are saved in a database and are retrieved whenever and wherever needed. In the future the system may be designed as a mobile application

**Future Enhancement**

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

* A printer can be added in the future for printing the bills.
* The platform can be hosted on the online servers to make it accessible worldwide.
* Integrate multiple load balancers to distribute the loads of the system.
* Implement backup mechanism for taking backup of database on regular basis on different servers.
* Email or message can be sent to the customer once the status is set to ready.
* Online payment gateway can be added like Paytm, Google Pay and phonepe.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of customers and their orders. I have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

**BIBLIOGRAPHY**

**Web References**

1. <https://www.w3schools.in/dbms/intro/>

Introduction to DBMS

2. <https://www.w3schools.com/sql/sql_intro.asp>

Introduction to SQL

3. <https://dev.mysql.com/doc/>

MySQL documentation

4. <https://dev.mysql.com/doc/>

PHP documentation

5. <https://dev.mysql.com/doc/refman/8.0/en/stored-routines.html>

Stored Procedures documentation

6. [https://stackoverflow.com/](https://stackoverflow.com/%20)

Stack overflow

7. <https://www.coursera.org/learn/database-applications-php>

Database applications in php

8. <https://www.coursera.org/learn/web-applications-php>

Web applications in php

**Book References**

1. Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition,

2017, Pearson

**YouTube References**

1.<https://youtu.be/dwVj_g3TpZ4>

2. <https://youtu.be/V_lAhqLXT9A>

3. <https://youtube.com/c/ProgrammingwithVishal>