MY CART

A Major Project II

Submitted in partial fulfillment of the requirements
for the degree of

BACHELOR OF TECHNOLOGY

(Computer Science & Engineering)

by

Aman Kumar Vishwakarma
0225CS191010

Ram Sujan Rajak
0225CS191046

Nikhil Raj Sahu
0225CS191036

Shailendra Shrivastav
0225CS191054

Under the guidance of

Prof. Abhishek Singh

Department of Computer Science & Engineering

Baderia Global Institute Of Engineering & Management, Jabalpur (M.P.)

Under

RAJIV GANDHI PRODYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)

APR-2023

Baderia Global Institute Of Engineering & Management, Jabalpur(m,p)

Department of Computer Science & Engineering

Certificate of HOD & Guide

This is to certify that the Major Project report entitled "My Cart" submitted by Aman Kumar Vishwakarma, Nikhil Raj Sahu, Shailendra Shrivastav and Ram Sujan Rajak has been carried out under my guidance & supervision. The project report is approved for submission towards partial fulfillment of the requirement for the award of degree of **Bachelor of Engineering** in **Computer Science & Engineering** from "**Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P).**

Prof. Abhishek Singh
Project Guide

Prof. Saurabh Sharma
HOD
Dept of CSE

Baderia Global Institute Of Engineering & Management, Jabalpur (M.P.)

Department of Computer Science & Engineering

Certificate of Internal & External Examiner

This is to certify that the Major Project report entitled "**MY CART**" is submitted by Aman Kumar Vishwakarma, Nikhil Raj Sahu, Shailendra Shrivastav and Ram Sujan Rajak, for the partial fulfillment of the requirement for the award of degree of **Bachelor of Engineering** in **Computer Science & Engineering** from **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P).**

Internal Examiner
Date :

External Examiner
Date :

ABSTRACT

Lack of Security:

The paper document is less secure compared to an electronic system. Misplaced documents can easily get into the wrong hands. The organization Secret or classified information is unsafe. If you lost them somewhere, there is no chance of getting them back. The leak of confidential documents can cost you a big amount. Clients expect their information to be secure in your hands. If you can't keep this safe, you are at risk of losing them.

Time Consuming:

Manually managing is a very tough and time-consuming process. Handling each and every document and storing them safely is not easy. It's not only about security but also about transporting or carrying out documents at different places. Physically carrying the documents and handling them is a task. But if you are using an IT tool then you can travel with years of data in just an eraser size pen-drive.

Insufficient Storage Space:

Paperwork can take up a significant amount of space, and the need for space will increase as the number of the document increases with an increase in growth of the business. The business increase, paperwork also increases, data will increase, records will also increase leads to insufficient space.

Difficulty in a modification of data:

When you are working with paper documents, it is much harder to make changes. If you want to make any change you will have to make a copy, so you don't destroy the original with any edits or comments you might add. This means the editing process is more time consuming and costly than if you were working with digital copies.

Offline Business is Expensive:

Compared to online business, offline business is more expensive since it requires maintaining a physical store which may take a huge amount of revenue as well as requires time. Offline business also requires employees.

Searching Product:

Searching a particular product in an offline store is quite difficult for both the staff as well as for the customer. Compared to this in online business searching for a particular product is quite easy and fast.

Decrease in Demand:

In offline business the owner buys the product on the basis of predictions and guessing instead of market trends. It can lead to loss if the demand for that product is less.

Declaration

I / We hereby declare that the project entitled **"MY CART"** which is being submitted in partial fulfillment of the requirement for award of the Degree of Bachelor of Engineering in Computer Science and Engineering to **"RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)"** is an authentic record of our own work done under the guidance of SAURABH KAPOOR, Department Computer Science & Engineering, **GLOBAL ENGINEERING COLLEGE, JABALPUR**.

The matter reported in this Project has not been submitted earlier for the award of any other degree.

Aman Kumar Vishwakarma
0225CS191010

Ram Sujan Rajak
0225CS191046

Nikhil Raj Sahu
0225CS191036

Shailendra Shrivastav
0225CS191054

Acknowledgment

We sincerely express indebtedness to esteemed and revered guide “Abhishek Singh”, in the department of CSE for his invaluable guidance, supervision and encouragement throughout the work. Without his kind patronage and guidance the project would not have taken shape.

We take this opportunity to express a deep sense of gratitude to Prof. Abhishek Singh professor of “Department of Computer Science & Engineering” for his encouragement and kind approval. Also we thank him for providing the computer lab facility. We would like to express our sincere regards to him for advice and counseling from time to time.

We owe sincere thanks to all the lecturers in “Department of Computer Science & Engineering*”* for their advice and counseling from time to time.

Aman Kumar Vishwakarma
DATED:
0225CS191010
PLACE:

Ram Sujan Rajak
0225CS191046

Nikhil Raj Sahu
0225CS191036

Shailendra Shrivastav
0225CS191054

INDEX

Chapter 1: Introduction

OBJECTIVE OF THE PROJECT

The main objective of this project is to make it easier to work through inventory on a daily basis. This is very much required and highly significant in the management of inventory. This convenient system of working will take away all the major concerns and helps in smooth and effective functioning of a business. Most importantly, it takes very less time to navigate through the software before beginning to work. It is more prominent and more significant in helping businesses to stay ahead and be successful in business. There is every accounting solution that can provide reliable service apart from keeping the safety and security of all business data.

Functionalities provided by Shop Management System are as follows:

Provides the searching facilities based on various factors. Such as Products, Payments, client, date of purchase.

Shop Management System also manages the Inventory details like date of purchase, quantity, rate, of products. It tracks all the information of sales, Inventory, Discounts etc. Manage the information of sales, Shows the information and description of the Products, payments, to increase efficiency of managing the Products, sales. It deals with monitoring the information and transactions of payments. Manage the information of Products, editing, adding and updating of Records is improved which results in proper. Resource management of Products data.

PROBLEM DEFINITION

The problem behind making this software was achieving an effective yet simple solution for the problems faced by vendors in achieving an effective way to manage their sales records. Managing paper records is a tough task, sometimes if misplaced it might lead to discrepancies and create problems. manually written documents can be tempered leading to wrong information. transaction related documents could be misplaced or tampered with, leading to loss in business. Searching information in papers is a hectic task, resulting in wastage of time. keeping papers for a long period of time might cause shattering A shopkeeper has to store records i.e. rate, quantity, date of purchasing, seller name, buyer name etc. of every product bought and sold. There are various software available in the market to achieve the desirable outcomes but either they are very expensive, or complex, or require a high level machine to work. We designed a standalone software to encounter this problem with minimal requirements. This will help shopkeepers to manage and run their business smoothly.

Chapter 2:

Feasibility study

Feasibility Study

After doing the project ‘my cart’, study and analysing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

A. Economic Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor. All hardware and software cost has to be borne by the organization. Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for the system.

B. Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if every was possible using different type of frontend and backend platforms.

C. Operational Feasibility

No doubt the proposed system is fully GUI based and is very user friendly and all inputs to be taken are all self-explanatory even to a layman. Besides, proper

training has been conducted to let them know the essence of the system so that they feel comfortable with the new system. As far as our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

D. Benefit to Organization

The organization will obviously be able to gain benefits such as savings in operating cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill.

E. The Initial Cost

The initial cost of setting up the system will include the cost of hardware software (OS, add-on software, utilities) & labour (setup & maintenance). The same has to be borne by the organization.

Chapter 3: Project Planning and Schedule

Project Planning and Schedule

The Project is designed and developed for business holders so that they can do business easily. Our planning started when we found the flaw in system., In the existing system or pen -paper mode system, there are so many discrepancies present like they cannot be stored for a long time . Pen paper mode written bills can be tempered easily and this can create a problem and sometimes due to misunderstanding the business holders cause a huge loss. So we have thought about the problem and we planned to develop a software which can deal with this problem.

Planning

After understanding the problem, we planned to develop a software. Our planning is to develop a software which can deal with all discrepancies as well as it should be easy to use for business holders so we developed a software with the help of Java Swing Language for User Interface and we have used Wamp Server for database and for developing a software in Java Language we used Eclipse Platform, we found it pretty good for developing a software in Java Swing and we also used Core Java for developing a Software

Schedule

We scheduled this software to develop as soon as possible. so, we planned to do it in a 3-month and we have learnt day-by-day and we have improved. Now we have developed a software that can deal with all the vulnerabilities of existing systems.

Chapter 4:

Technology used

TECHNOLOGY USED

FRONTEND

1.HTML(HyperText Markup Language):-

The HyperText Markup Language or 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.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[2] A form of HTML, known as HTML5, is used to display video and audio, primarily using the <**canvas**> element, in collaboration with javascript.

2.CSS(Cascading Style Sheets):-

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.[2]

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.[4]

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents

3. JAVASCRIPT:-

JavaScript often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.[11] Over 97% of websites use JavaScript on the client side for web page behavior,[12] often incorporating

third-party libraries.[13] All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[14] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

4. BOOTSTRAP:-

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface

elements, including for example an auto-complete function for input fields.

Example of a webpage using Bootstrap framework rendered in Firefox

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the five predefined fixed widths, depending on the size of the screen showing the page:

- Smaller than 576 pixels
- 576–768 pixels
- 768–992 pixels
- 992–1200 pixels
- Larger than 1200 pixels

Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.

A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

BACKEND

1. DJANGO:-

Django is a widely-used Python web application framework with a "batteries-included" philosophy. The principle behind batteries-included is that the common functionality for building web applications should come with the framework instead of as separate libraries.

For example, authentication, URL routing, a template engine, an object-relational mapper (ORM), and database schema migrations are all included with the Django framework. Compare that included functionality to the Flask framework which requires a separate library such as Flask-Login to perform user authentication.

The batteries-included and extensibility philosophies are simply two different ways to tackle framework building. Neither philosophy is inherently better than the other one.

The Django project's stability, performance and community have grown tremendously over the past decade since the framework's creation. Detailed tutorials and good practices are readily available on the web and in books. The framework continues to add significant new functionality such as database migrations with each release.

I highly recommend the Django framework as a starting place for new Python web developers because the official documentation and tutorials are some of the best anywhere in software development. Many cities also have Django-specific groups such as Django District, Django Boston and San Francisco Django so new developers can get help when they are stuck

2. SQLITE3:-

SQLite is a self-contained, file-based SQL database. SQLite comes bundled with Python and can be used in any of your Python applications without having to install any additional software.

In this tutorial, we'll go through the sqlite3 module in Python 3. We'll create a connection to a SQLite database, add a table to that database, insert data into that table, and read and modify data in that table.

For this tutorial, we'll be working primarily with an inventory of fish that we need to modify as fish are added to or removed from a fictional aquarium.

3. PYTHON:-

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.[30]

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.[31][32]

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0.[33] Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.[34]

Python consistently ranks as one of the most popular programming languages.

Chapter 5

Module Description

Module Descriptions

We have 7 modules they are as follows:

Module 1: Customers

In this module we will talk about what our software will do in Customer section, when customer option is clicked a new window will open in which 5 options will open they are as follows:

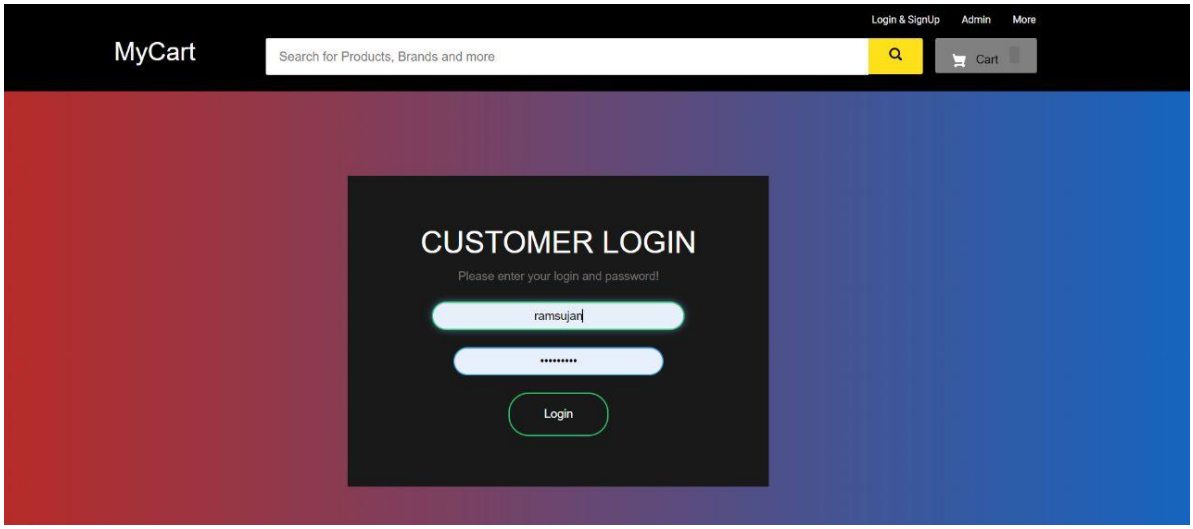
Customer Id

Customer Name

Customer Address

Customer Mobile

Date of Addition

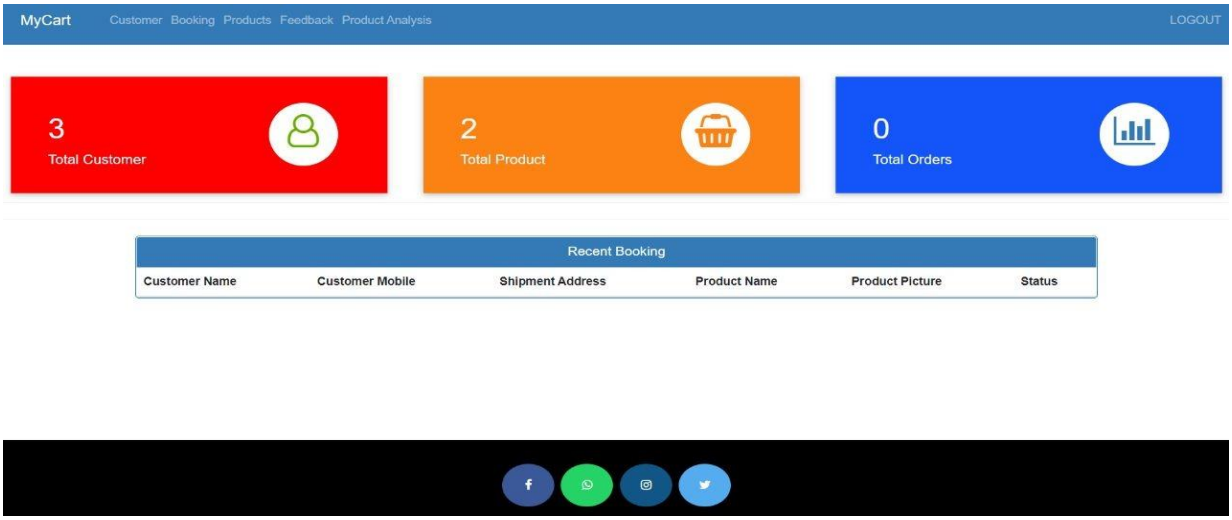


If a Customer is new then new Customer Id is assigned to him if Customer is already saved in database then customer id is filled with the given number then we have second option Customer Name in which customer name is saved if the customer is new then the Customer name is saved in database , if existed customer is present then the name is filled automatically that's our upcoming plan then another option is Customer Address it is also same as previous one if customer is new then it will save, if existed then no problem will happen same with other option customer mobile . Now we have another good option date of addition this will store date when record is added of Customer

Module 2: Admin

In this module we will talk about what our software will do in admin section, when admin option is clicked a new window will open in which 5 options will open, they are as follows:

- Customer
- Booking
- Products
- Feedback
- Product Analysis



If a product is new then new product id is assigned to him if Supplier is already saved in database then supplier id is filled with the given number then we have second option Supplier Name in which supplier name is saved if the supplier is new then the Supplier name is saved in database , if existed customer is present then the name is filled automatically that's our upcoming plan then another option is Supplier Address it is also same as previous one if supplier is new then it will save, if existed then no problem will happen same with other option supplier mobile . Now we have another good option: the date of addition this will store the date when the record is added to the Supplier.

Module 3: Purchases

In this module we will talk about what our software will do in Purchases section, when supplier option is clicked a new window will open in which 10 options will open, they are as follows:

Serial Number

Purchase Id

Item Name

Item Quantity

Item Price

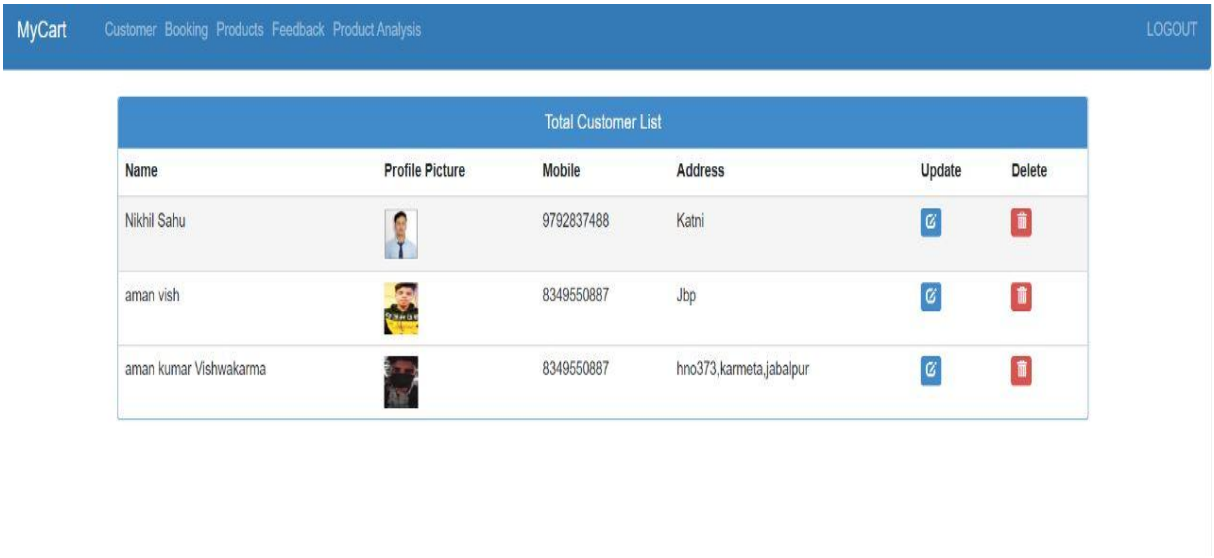
Labour Charges

Freight

Balance Quantity

Date of Purchases

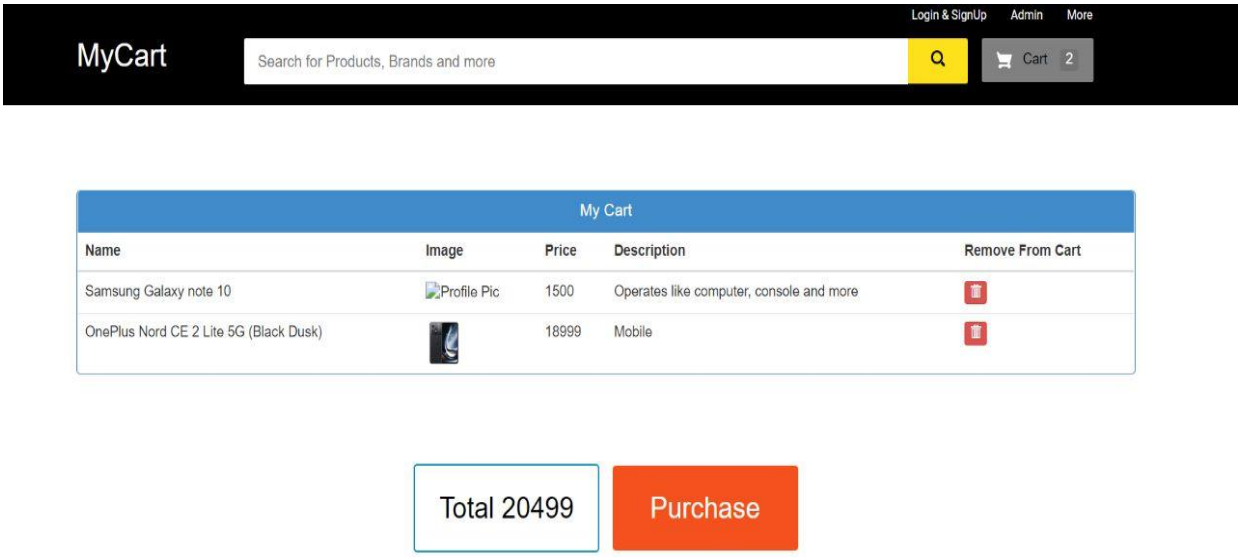
Date of Addition



In this section Item serial number is stored systematically first then supplier id is stored then item name, item quantity, item price, labour charges, freight, balance quantity, date of purchases and date of addition Note date of purchases and date of addition can be different

Module 4: Cart

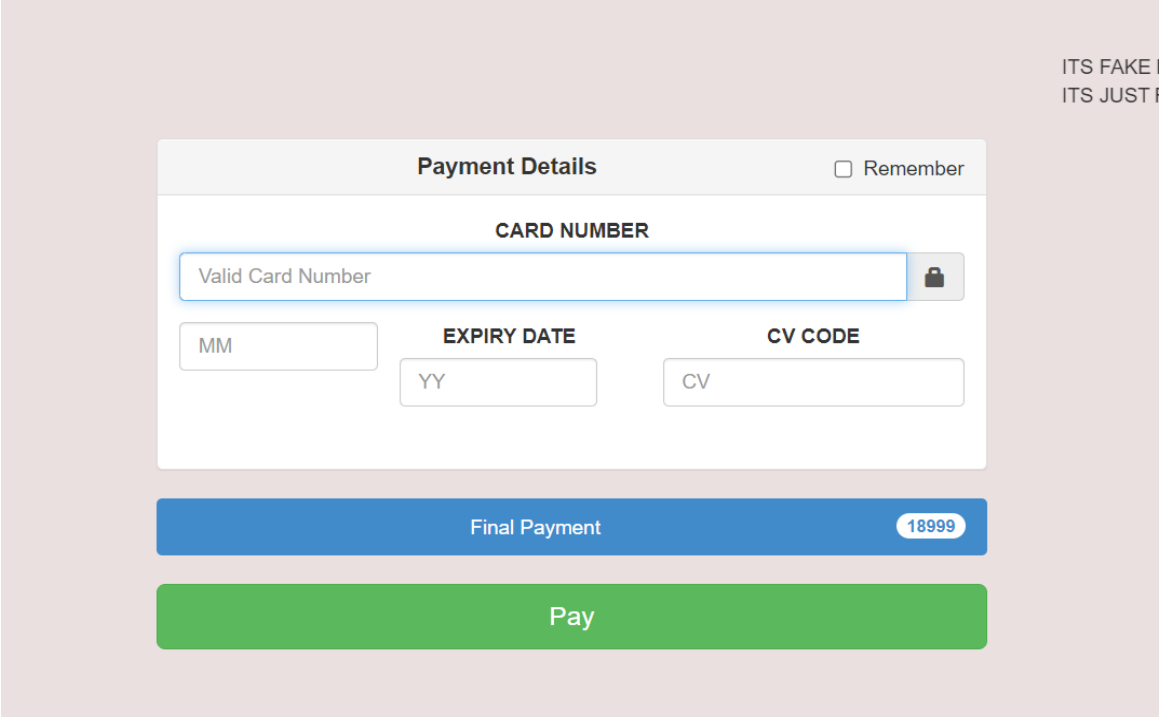
In this module we will talk about what our software will do in Cart section, when customer likes any product then he/she can add it into the cart and then buy later.



In this section Item serial number is stored systematically first then customer id is stored then item name, item quantity, item price, labour charges, freight, balance quantity, date of purchases and date of addition Note date of purchases and date of addition can be different

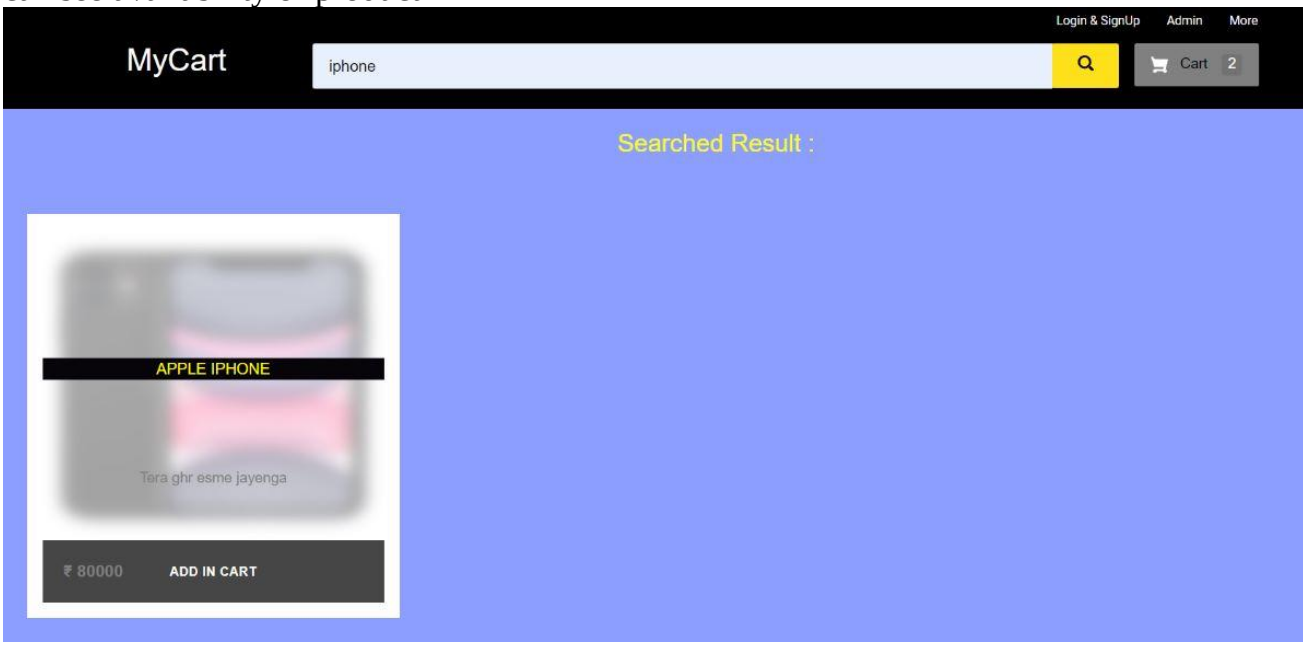
Module 5: PAYMENT

In this section bill is printed that is generated ii above sections and it will print the bills that's in our upcoming plan



Module 6: Search

In this section Search option is provided. We can search Prroduct, Coustomers can see availability of product



Module 7: Inventories management

A company's inventory is one of its most valuable assets. In retail, manufacturing, food services, and other inventory-intensive sectors, a company's inputs and finished products are the core of its business. A shortage of inventory when and where it's needed can be extremely detrimental.

At the same time, inventory can be thought of as a liability (if not in an accounting sense). A large inventory carries the risk of spoilage, theft, damage, or shifts in demand. Inventory must be insured, and if it is not sold in time it may have to be disposed of at clearance prices—or simply destroyed.

Chapter 6

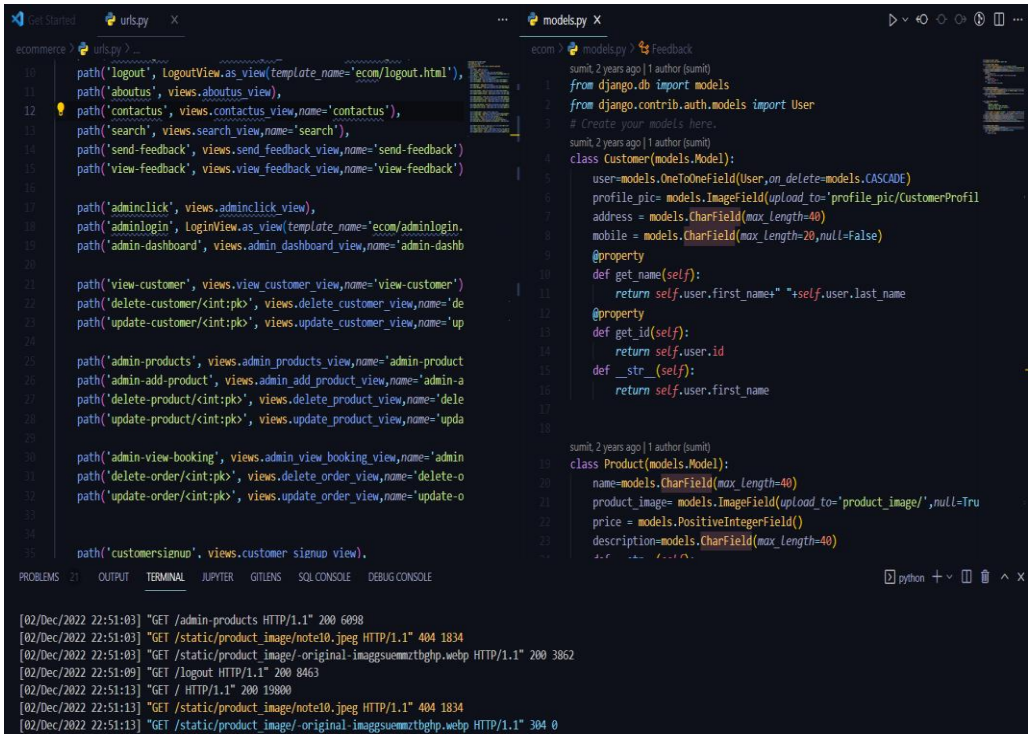
Working Code

Database Table / Data File Structure

A well-structured database: Saves disk space by eliminating redundant data. Maintains data accuracy and integrity. Provides access to the data in useful ways.

Designing an efficient, useful database is a matter of following the proper process, including these phases: Requirement's analysis, or identifying the purpose of your database

Organizing data into tables Specifying primary keys and analysing relationships Normalizing to standardize the tables Let's take a closer look at

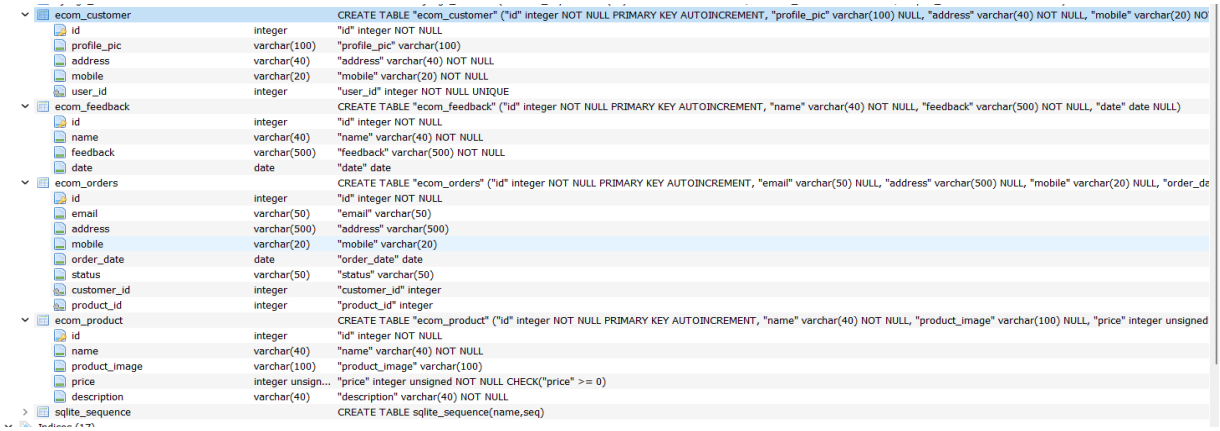


each step. Note that this guide deals with Edgar Codd's relational database model as written in SQL (rather than the hierarchical, network, or object data models). To learn more about database models, read our guide here.

Requirement's analysis: identifying the purpose of the database

Understanding the purpose of your database will inform your choices throughout the design process. Make sure you consider the database from every perspective. For instance, if you were making a database for a public library,

you'd want to consider the ways in which both patrons and librarians would need to access the data.



ecom_customer		CREATE TABLE "ecom_customer" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "profile_pic" varchar(100) NULL, "address" varchar(40) NOT NULL, "mobile" varchar(20) NO
id	integer	"id" integer NOT NULL
profile_pic	varchar(100)	"profile_pic" varchar(100)
address	varchar(40)	"address" varchar(40) NOT NULL
mobile	varchar(20)	"mobile" varchar(20) NOT NULL
user_id	integer	"user_id" integer NOT NULL UNIQUE
ecom_feedback		CREATE TABLE "ecom_feedback" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(40) NOT NULL, "feedback" varchar(500) NOT NULL, "date" date NULL)
id	integer	"id" integer NOT NULL
name	varchar(40)	"name" varchar(40) NOT NULL
feedback	varchar(500)	"feedback" varchar(500) NOT NULL
date	date	"date" date
ecom_orders		CREATE TABLE "ecom_orders" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "email" varchar(50) NULL, "address" varchar(500) NULL, "mobile" varchar(20) NULL, "order_da
id	integer	"id" integer NOT NULL
email	varchar(50)	"email" varchar(50)
address	varchar(500)	"address" varchar(500)
mobile	varchar(20)	"mobile" varchar(20)
order_date	date	"order_date" date
status	varchar(50)	"status" varchar(50)
customer_id	integer	"customer_id" integer
product_id	integer	"product_id" integer
ecom_product		CREATE TABLE "ecom_product" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(40) NOT NULL, "product_image" varchar(100) NULL, "price" integer unsigned
id	integer	"id" integer NOT NULL
name	varchar(40)	"name" varchar(40) NOT NULL
product_image	varchar(100)	"product_image" varchar(100)
price	integer unsign...	"price" integer unsigned NOT NULL CHECK("price" >= 0)
description	varchar(40)	"description" varchar(40) NOT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)

Here are some ways to gather information before creating the database:

Interview the people who will use it Analyze business forms, such as invoices, timesheets, surveys

Comb through any existing data systems (including physical and digital files)

Start by gathering any existing data that will be included in the database. Then list the types of data you want to store and the entities, or people, things, locations, and events, that those data describe, like this:

Customers

Name

Address

City, State, Zip

Email address

Products

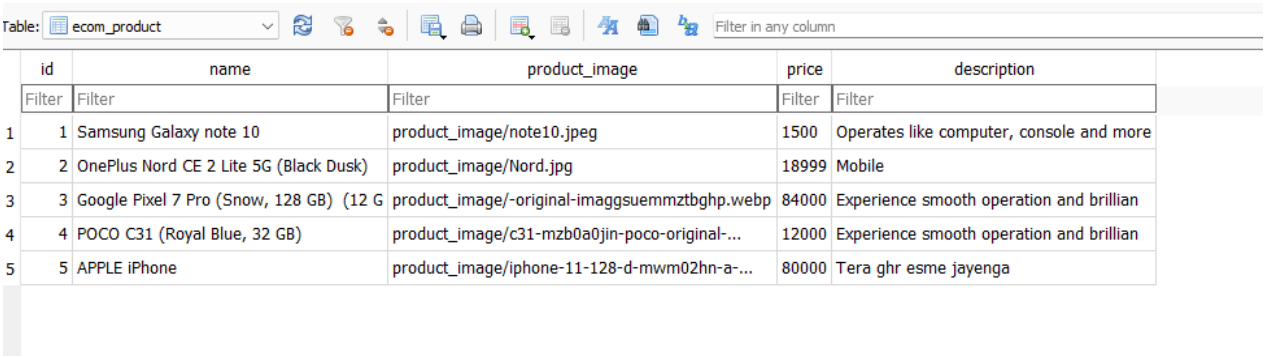


Table: ecom_product — Filter in any column

	id	name	product_image	price	description
	Filter	Filter	Filter	Filter	Filter
1	1	Samsung Galaxy note 10	product_image/note10.jpeg	1500	Operates like computer, console and more
2	2	OnePlus Nord CE 2 Lite 5G (Black Dusk)	product_image/Nord.jpg	18999	Mobile
3	3	Google Pixel 7 Pro (Snow, 128 GB) (12 G	product_image/-original-imaggsuemmztbghp.webp	84000	Experience smooth operation and brillian
4	4	POCO C31 (Royal Blue, 32 GB)	product_image/c31-mzb0a0jin-poco-original-...	12000	Experience smooth operation and brillian
5	5	APPLE iPhone	product_image/iphone-11-128-d-mwm02hn-a-...	80000	Tera ghr esme jayenga

Name

Price

Quantity in stock

Quantity on order

Orders
Order ID
Sales representative
Date
Product(s)
Quantity
Price
Total
This information will later become part of the data dictionary, which outlines the tables and fields within the database. Be sure to break down the information into the smallest useful pieces. For instance, consider separating the street address from the country so that you can later filter individuals by their country of residence. Also, avoid placing the same data point in more than one table, which adds unnecessary complexity. Once you know what kinds of data the database will include, where that data comes from, and how it will be used, you're ready to start planning out the actual database. Database structure: the building blocks of a database
The next step is to lay out a visual representation of your database. To do that, you need to understand exactly how relational databases are structured. Within a database, related data are grouped into tables, each of which consists of rows (also called tuples) and columns, like a spreadsheet.

To convert your lists of data into tables, start by creating a table for each type of entity, such as products, sales, customers, and orders. Here's an example: Each row of a table is called a record. Records include data about something or someone, such as a particular customer. By contrast, columns (also known as fields or attributes) contain a single type of information that appears in each record, such as the addresses of all the customers listed in the table.

First Name	Last Name	Age	ZIP Code
Roger	Williams	43	34760
Jerrica	Jorgensen	32	97453
Samantha	Hopkins	56	64829

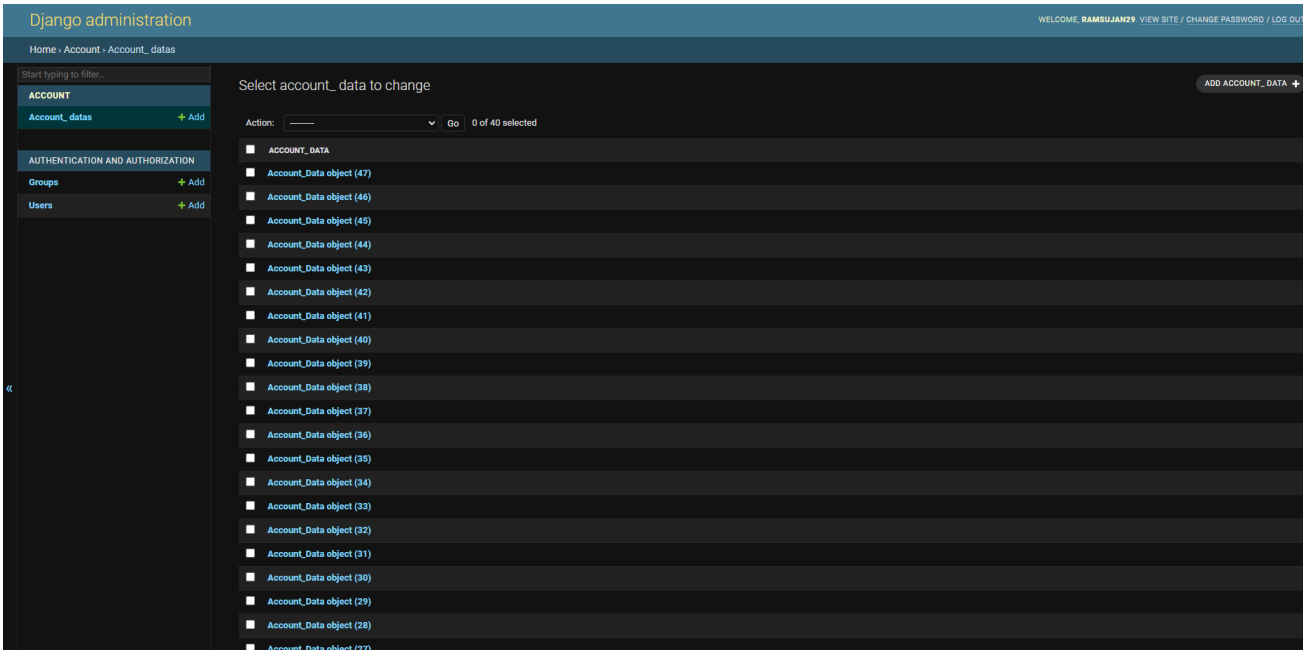
To keep the data consistent from one record to the next, assign the appropriate data type to each column. Common data types include:
CHAR - a specific length of text
VARCHAR - text of variable lengths
TEXT - large amounts of text

INT - positive or negative whole number
FLOAT, DOUBLE - can also store floating point numbers
BLOB - binary data
Some database management systems also offer the Auto number data type, which automatically generates a unique number in each row.
For the purposes of creating a visual overview of the database, known as an entity-relationship diagram, you won't include the actual tables. Instead, each table becomes a box in the diagram. The title of each box should indicate what the data in that table describes, while attributes are listed below, like this:

Chapter 7

DATA BASE

Database table



Finally, you should decide which attribute or attributes will serve as the primary key for each table, if any. A primary key (PK) is a unique identifier for a given entity, meaning that you could pick out an exact customer even if you only knew that value. Attributes chosen as primary keys should be unique, unchanging, and always present (never NULL or empty). For this reason, order numbers and usernames make good primary keys, while telephone numbers or street addresses do not. You can also use multiple fields in conjunction as the primary key (this is known as a composite key).

When it comes time to create the actual database, you'll put both the logical data structure and the physical data structure into the data definition language supported by your database management system. At that point, you should also estimate the size of the database to be sure you can get the performance level and storage space it will require.

Creating relationships between entities

With your database tables now converted into tables, you're ready to analyse the relationships between those tables. Cardinality refers to the quantity of elements that interact between two related tables. Identifying the cardinality helps make sure you've divided the data into tables most efficiently.Each entity can potentially have a relationship with every other one, but those relationships are typically one of three types:

One-to-one relationships

When there's only one instance of Entity A for every instance of Entity B, they are said to have a one-to-one relationship (often written 1:1). You can indicate this kind of relationship in an ER diagram with a line with a dash on each end: Unless you have a good reason not to, a 1:1 relationship usually indicates that you'd be better off combining the two tables' data into a single table.

However, you might want to create tables with a 1:1 relationship under a particular set of circumstances. If you have a field with optional data, such as "description," that is blank for many of the records, you can move all of the descriptions into their own table, eliminating empty space and improving database performance.

To guarantee that the data matches up correctly, you'd then have to include at least one identical column in each table, most likely the primary key.

One-to-many relationships

These relationships occur when a record in one table is associated with multiple entries in another. For example, a single customer might have placed many orders, or a patron may have multiple books checked out from the library at once. One-to-many (1:M) relationships are indicated with what's called "Crow's foot notation," as in this example: one to many relationship

To implement a 1:M relationship as you set up a database, simply add the primary key from the "one" side of the relationship as an attribute in the other table. When a primary key is listed in another table in this manner, it's called a

foreign key. The table on the “1” side of the relationship is considered a parent table to the child table on the other side.

Many-to-many relationships

When multiple entities from a table can be associated with multiple entities in another table, they are said to have a many-to-many (M: N) relationship. This might happen in the case of students and classes, since a student can take many classes and a class can have many students.

In an ER diagram, these relationships are portrayed with these lines:

Many to many relationship

Unfortunately, it’s not directly possible to implement this kind of relationship in a database. Instead, you have to break it up into two one-to-many relationships. To do so, create a new entity between those two tables. If the M:N relationship exists between sales and products, you might call that new entity “sold products,” since it would show the contents of each sale. Both the sales and products tables would have a 1:M relationship with sold products. This kind of go-between entity is called a link table, associative entity, or junction table in various models.

Each record in the link table would match together two of the entities in the neighbouring tables (it may include supplemental information as well). For instance, a link table between students and classes might look like this:

link table

Mandatory or not?

Another way to analyse relationships is to consider which side of the relationship has to exist for the other to exist. The non-mandatory side can be marked with a circle on the line where a dash would be. For instance, a country has to exist for it to have a representative in the United Nations, but the opposite is not true:

mandatory or not

Two entities can be mutually dependent (one could not exist without the other).

Recursive relationships

Sometimes a table points back to itself. For example, a table of employees might have an attribute “manager” that refers to another individual in that same table. This is called a recursive relationship.

Redundant relationships

A redundant relationship is one that is expressed more than once. Typically, you can remove one of the relationships without losing any important information. For instance, if an entity “students” has a direct relationship with another called “teachers” but also has a relationship with teachers indirectly through “classes,” you’d want to remove the relationship between “students” and “teachers.” It’s better to delete that relationship because the only way that students are assigned to teachers is through classes.

7.1

Database normalization

Once you have a preliminary design for your database, you can apply normalization rules to make sure the tables are structured correctly. Think of these rules as the industry standards.
That said, not all databases are good candidates for normalization. In general, online transaction processing (OLTP for short) databases, in which users are concerned with creating, reading, updating, and deleting records, should be normalized.
Online analytical processing (OLAP) databases which favour analysis and reporting might fare better with a degree of denormalization, since the emphasis is on speed of calculation. These include decision support applications in which data needs to be analysed quickly but not changed.
Each form, or level of normalization, includes the rules associated with the lower forms.

First normal form

The first normal form (abbreviated as 1NF) specifies that each cell in the table can have only one value, never a list of values, so a table like this does not comply:

Productid	Colour	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

You might be tempted to get around this by splitting that data into additional columns, but that's also against the rules: a table with groups of repeated or closely related attributes does not meet the first normal form. The table below, for example, fails to comply:
Instead, split the data into multiple tables or records until each cell holds only one value and there are no extra columns. At that point, the data is said to be atomic, or broken down to the smallest useful size. For the table above, you could create an additional table called "Sales details" that would match specific products with sales. "Sales" would then have a 1:M relationship with "Sales details."

Second normal form

The second normal form (2NF) mandates that each of the attributes should be fully dependent on the entire primary key. That means each attribute should depend directly on the primary key, rather than indirectly through some other attribute.
For instance, an attribute "age" that depends on "birthdate" which in turn depends on "studentID" is said to have a partial functional dependency, and a table containing these attributes would fail to meet the second normal form.
Furthermore, a table with a primary key made up of multiple fields violates the second normal form if one or more of the other fields do not depend on every part of the key.
Thus, a table with these fields wouldn't meet the second normal form, because the attribute "product name" depends on the product ID but not on the order number:

Order number (primary key)
Product ID (primary key)
Product name

Third normal form

The third normal form (3NF) adds to these rules the requirement that every non-key column be independent of every other column. If changing a value in

one non-key column causes another value to change, that table does not meet the third normal form.

This keeps you from storing any derived data in the table, such as the “tax” column below, which directly depends on the total price of the order:

Order Price Tax

14325 $40.99$2.05

14326 13.73.69

14327 $24.15$1.21

Additional forms of normalization have been proposed, including the Boyce-Codd normal form, the fourth through sixth normal forms, and the domain-key normal form, but the first three are the most common.

While these forms explain the best practices to follow generally, the degree of normalization depends on the context of the database.

Diagramming is quick and easy with Lucid chart. Start a free trial today to start creating and collaborating.

Make a database diagram

Multidimensional data

Some users may want to access multiple dimensions of a single type of data, particularly in OLAP databases. For instance, they may want to know the sales by customer, state, and month. In this situation, it’s best to create a central fact table that other customer, state, and month tables can refer to, like this: multidimensional database Data integrity rules

You should also configure your database to validate the data according to the appropriate rules. Many database management systems, such as Microsoft Access, enforce some of these rules automatically. The entity integrity rule says that the primary key can never be NULL. If the key is made up of multiple columns, none of them can be NULL. Otherwise, it could fail to uniquely identify the record.

The referential integrity rule requires each foreign key listed in one table to be matched with one primary key in the table it references. If the primary key changes or is deleted, those changes will need to be implemented wherever that key is referenced throughout the database.

Business logic integrity rules make sure that the data fits within certain logical parameters. For instance, an appointment time would have to fall within normal

business hours. Adding indexes and views An index is essentially a sorted copy of one or more columns, with the values either in ascending or descending order. Adding an index allows users to find records more quickly. Instead of re-sorting for each query, the system can access records in the order specified by the index.

Although indexes speed up data retrieval, they can slow down inserting, updating, and deleting, since the index has to be rebuilt whenever a record is changed.

A view is simply a saved query on the data. They can usefully join data from multiple tables or else show part of a table.

Extended properties

Once you have the basic layout completed, you can refine the database with extended properties, such as instructional text, input masks, and formatting rules that apply to a particular schema, view, or column. The advantage is that, because these rules are stored in the database itself, the presentation of the data will be consistent across the multiple programs that access the data.

Chapter 8

CONCLUSION

Conclusion

This web application will provide a one place solution for business management. It will help users in many ways to expand business. It will reduce pen and paper work. Records will be maintained for a long period of time without any discrepancies. Inventory stocking will decrease. It is more prominent and more significant in helping businesses to stay ahead and be successful in business. There is every accounting solution that can provide reliable service apart from keeping the safety and security of all business data. Further it also offers a lot of flexibility, easy access and is completely secured in many ways.

CHAPTER 9

FUTURE SCOPE

Future Scope and Further

Enhancement of the Project

We can add printers in the future. We can give more advance software for Shop Management System including more facilities We will host the platform on online servers to make it accessible worldwide Integrate multiple load balancers to distribute the loads of the system • Create the master and slave database structure to reduce the overload of the database queries Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

REFERENCES

WE HAVE TAKEN THE REFERENCES FOR JAVA FROM
https://codeforwin.org/
https://www.w3schools.com/
https://javatutoring.com/
https://www.javatpoint.com/
https://en.wikipedia.org/wiki/HTML
REFERENCES FOR DATABASE DESIGN
https://www.lucidchart.com/pages/database-diagram/database-design