CHAPTER 1

INTRODUCTION

This chapter will discuss the various features and aim of this application.

1.1 Introduction to Zayka:

This chapter will discuss about the problems faced by most of the restaurant, the proposed solution with project aims and objectives to overcome the problems, targeted company for the new system, project scope, overview diagram and general description about the modules of the system, the system development environment, operational environment, system architecture, network architecture as well as the methodology practiced in the project development. This chapter also includes the information about project scheduling and project team organization.

1.2 Objective:

This is a web application that runs the restaurant service. Since the main problem faced by most of the restaurant is the inefficient and insufficient work force to serve a large number of customers, especially during peak hour. Therefore, this system serves to improve restaurant service and management by maintaining all the restaurant information, accept food order from the web application, efficiently arrange food preparing lists for kitchen, manage customer reservation and generate useful report.

The web application of this system is used to maintain restaurant data, manage reservation, perform food ordering, and reporting. Data maintenance includes membership and menu. While for the kitchen, system is featured with intelligent food order arrange to improve food preparing. Staff can mark the food status to served, preparing or queuing.

The aims and objective of this system are to:

- Reduce human workforce in the restaurant. Since some of the manual works is being computerized such as staff management, customer reservation and customer membership, therefore the needs of larger workforce is not necessary.
- Simplify food ordering process by allowing the customers to make their own food ordering through our developed restaurant web application.
- Allow the kitchen department to work in an efficient and effective environment

as the priority queue arrangement of food list is implemented in this system. Besides, the order taken by front desk can be clearly seen on the monitor and able to obtain the new orders in a shorter time.

- Allow the restaurant staff to **efficiently in maintaining the restaurant information** such as customer membership, and menu information. The staff in-charge just need to key in the details through our system and our system will provide information maintenance to be made in a simple and easy way.

CHAPTER 2

SOFTWARE ENVIRONMENT

This chapter will discuss the software environment which is used to refer to support the application.

2.1 Database Management System (DBMS):

DBMS is a collection of programs that enables users to create and maintain a database The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating and sharing databases among various users and applications. A Relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. Data in a single table represents a relation, from which the name of the database type comes. In typical solutions, tables may have additionally defined relationships with each other. In the relational model, each table schema must identify a column or group of columns, called the primary key, to uniquely identify each row. A relationship can then be established between each row in the table and a row in another table by creating a foreign key, a column or group of columns in one table that points to the primary key of another table.

2.1.1 Characteristics of Database Management Systems:

- Provides insulation between Programs and data, Data abstraction.
- Supports multiple views of the data.
- Self-describing nature.
- Keeps a tight control on data redundancy.
- Enforces user defined rules to ensure that
- Helps sharing of data and Multi-user transaction processing.

2.1.2 Advantages of using the DBMS approach:

- Controlling the redundancy.
- Restricting unauthorized access.
- Providing persistent storage for program objects.
- Providing storage structures for efficient query processing.
- Providing backup and recovery.

- Providing multiple users interfaces
- Representing complex relationships among data.
- Enforcing integrity constraints.

2.1.3 Functional Dependency:

The Functional Dependency denoted by X →Y, between two sets of attributes X and Y that are subsets of R species a constraint on the possible tuples that can form a relation stare r of R. The constraints is that for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$.

This means that the values of the Y component of a tuple in r depend on, or determined by the values of the X components. Alternatively the values of the X component of a tuple uniquely determine the values of the Y component. Consider the following schema,

EMP_PRJ

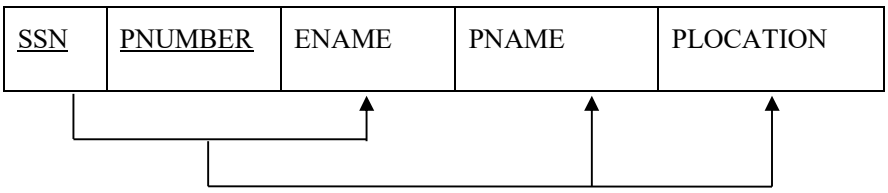


Fig 2.1

In the above schema (Fig 2.1) the functional dependencies are:

1. SSN → ENAME

   The value of an employee's social security number (SSN) uniquely determines the employee name (ENAME).

2. PNUMBER →{ PNAME, PLOCATION}

   These values of project's number uniquely determines the project name (PNAME) and project locations (PLOCATIONS)

2.1.4 Normalization:

The normalization process was proposed by Codd, it takes a relation schema through a series of tests to certify whether if satisfies a certain normal form. The process proceeds in a top down fashion by evaluating each relation against the criteria for normal forms and decomposing relations as necessary can thus be considered a relational design by analysis.

Normalization of data can be looked upon as a process of analyzing the given relation schemas based on their Functional Dependencies and primary keys to achieve the desirable properties of:

- Minimizing redundancy.
- Minimizing the insertion, deletion, and update anomalies.

2.1.5 Super key:

A super key of a relation schema R= {A_1, A_2… A_n} is asset of attributes S $\underline{C}$R with the property that no two tuples t_1 and t_2 in any legal relation state r of R will have $t_1[S] = t_2[S]$. A key K is a super key with the additional property that removal of any attribute from K will cause K not to be a super key anymore.

For example consider the following schema,

EMPLOYEE

SSN	ENAME	BDATE	ADDRESS	DNUM

Fig 2.2

Here (Fig 2.2), {SSN} is the primary for EMPLOYEE, where as {SSN}, {SSN, ENAME}, {SSN, ENAME, BDATE} and any set of attributes that includes SSN are called super keys.

If a relation schema has more than one key, each key is called **candidate key**. An attribute of relation schema R is called a **prime attribute** of R if it is a member of some candidate key of R. An attribute is called **nonprime** if it is not a prime attribute or it is not a member of any candidate.

2.1.6 Normal Forms:

There are three normal forms

- First normal form
- Second normal form
- Third normal form

These were proposed by Codd as a sequence to achieve the desirable state of 3NF relations by progressing through the intermediate states of 1Nf and 2NF if needed.

- **First Normal Form (1NF):**

It states that the domain of attribute must include only atomic be values and that the value of any attribute in a tuple must be a single value from the domain of the attribute. Hence 1NF disallows having a set of values a tuple of values or a combination of both as an attribute value for a single tuple.

Consider the following department relation schema,

DEPARTMENT

DNAME	DNUMBER	DMRSSN

Fig 2.3

where the primary keys is DNUMBER and suppose that we extend it by including the DLOCATIONS as shown in below Fig 2.4. We assume that each department can have a number of locations.

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	3333	{Delhi, Mumbai, Mysore}
Administration	4	4444	Chennai
Headquarters	1	4545	Bangalore

Fig 2.4

Above relation table is not in 1NF because DLOCATIONS is not an atomic attributes. It can be converted into 1NF by following methods:

- Remove the attribute DLOCATION that violates 1NF and place it in a separate relation DEPT_LOCATIONS as shown in Fig 2.5.
- Expand the key so that there will be a separate tuple in the original DEPARTMENT relation for each location of a DEPARTMENT as shown in Fig 2.6.

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Bangalore
4	Chennai
5	Delhi
5	Mumbai
5	Mysore

Fig 2.5

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	3333	Delhi
Research	5	3333	Mumbai
Research	5	3333	Mysore
Administration	4	4444	Chennai
Headquarters	1	4545	Bangalore

Fig 2.6

- **Second Normal Form (2NF):**

This normal form is based on the **full functional dependency**. A functional dependency X →Y is full functional dependency if removal for any attribute A from X means that dependency does not hold any more.

A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

The EMP_PROJ in fig 2.7a is in 1NF but not in 2NF. The non prime attribute ENAME violates 2NF because of FD3. The functional dependencies FD2 and FD3 make ENAME,

PNAME and PLOCATION partially dependent on the primary key {SSN, PNUMBER} of EMP_PROJ thus violating 2NF. The relational schema in Fig 2.7a can be second normalized into number of 2NF relations EP1, EP2 and EP3 as shown in the Fig 2.7b.

Fig 2.7a

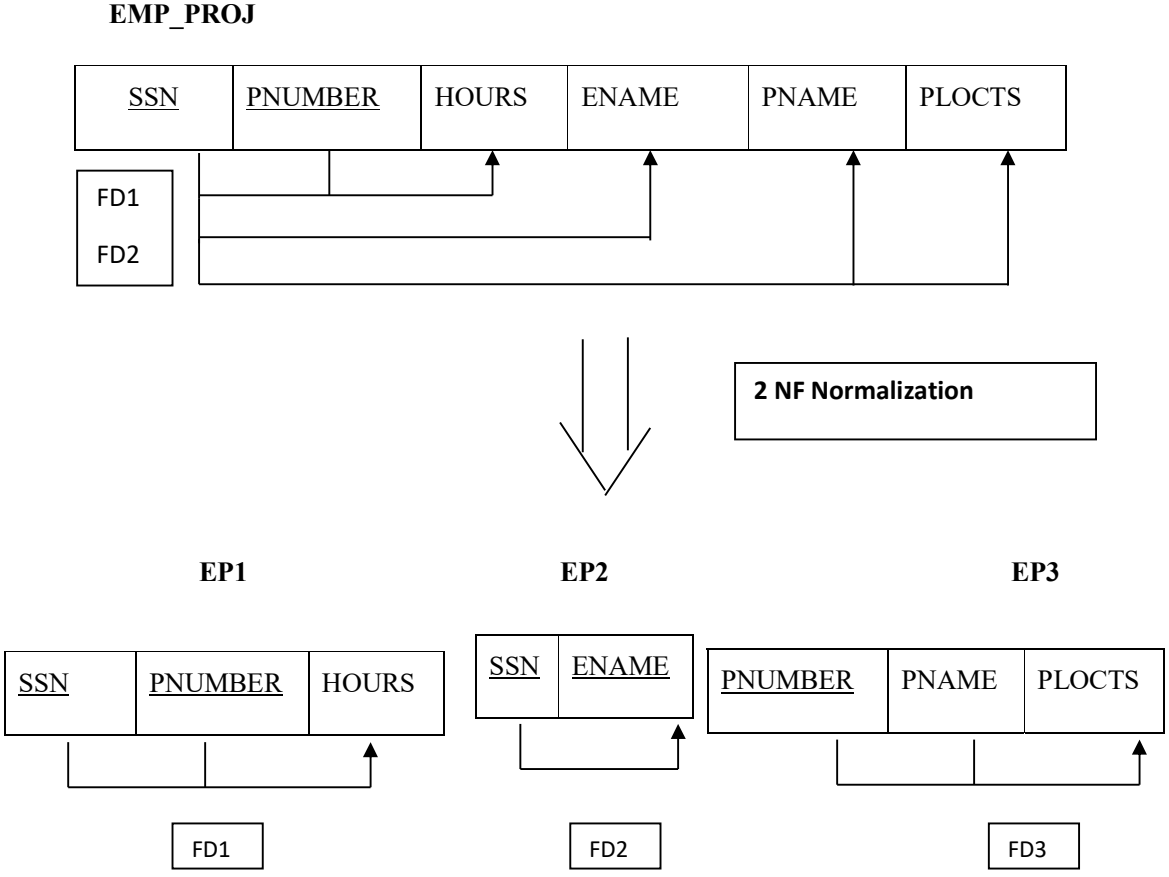


Fig 2.7b

Fig 2.7 Normalization of EMP_PROJ into 2NF

- **Third Normal Form:**

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.

A relation schema R is in 3Nf if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.

The relation schema EMP_DEPT in Fig 2.8a is in 2NF however it is not in 3NF because transitive dependency of DMRSSN on SSN via DNUMER. We can normalize EMP_DEPT by decomposing it into two 3NF relation schema ED1 and ED2 as shown in the Fig 2.8b.

Fig 2.8a

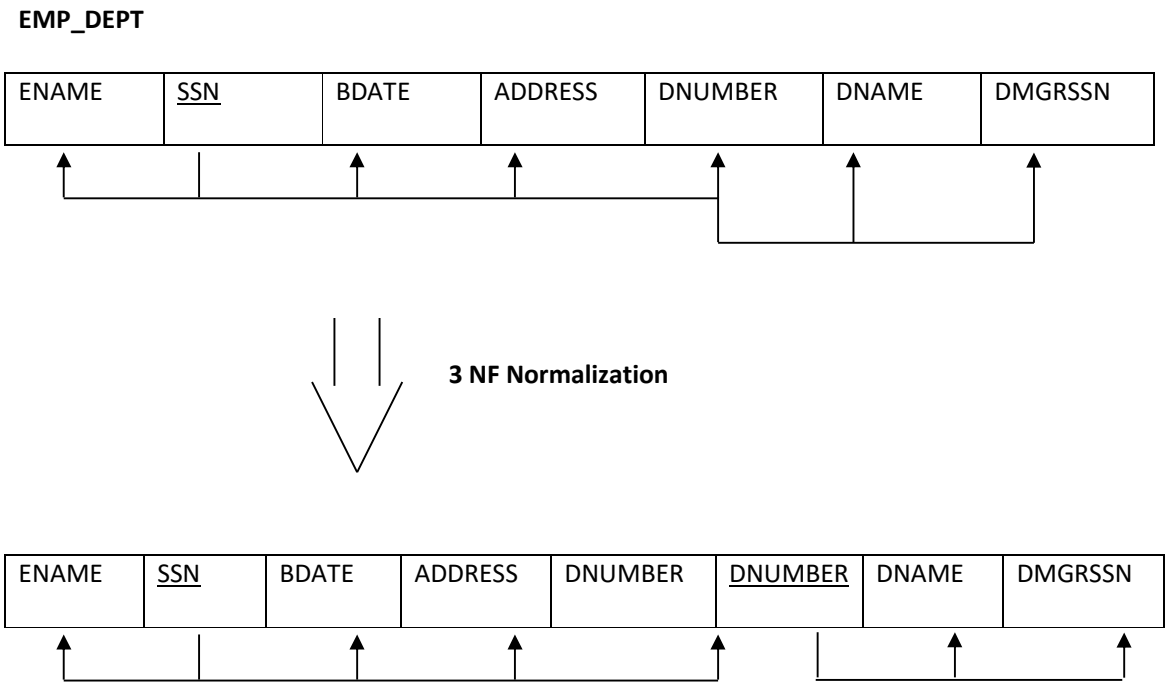


Fig 2.8b

Fig 2.8 Normalizing EMP_DEPT into 3NF relations

2.2 Structured Query Language (SQL):

The ANSI standard SQL provides basic functions for data manipulation, transaction control, and record retrieval from the database. However, most end users interact with Oracle through application that provides an interface that hides the underlying SQL and its complexity.

SQL uses the terms table, row, and column for relation, tuple, and attribute, respectively. The SQL commands for data definition are CREATE, ALTER and DROP.

- **CREATE:**

This command is used to create table or view by giving it a name and specifying its attributes and constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain values, and any attribute constraints such as NOT NULL.

SYNTAX: CREATE TABLE <TNAME> (ATR1 TYP1 CONST1, ATR2 TYP2 CONST,…)

- **ALTER:**

The definition of a base table can be altered by ALTER command which is a Schema Evolution command. The possible ALTER TABLE include adding or dropping a column (attribute), changing a column definition, and adding or dropping table constraints.

Example: ALTER TABLE STUDENT ADD NAME VARCHAR (12)

- **DROP:**

If a whole schema is not needed any more, the DROP SCHEMA command can be used. There are two drip behaviour options: CASCADE and RESTRICT.

CASCADE option is used to remove the database schema and all its tables, domains and other elements.

If the RESTRICT option is chosen in place of CASCADE, the schema is dropped only it has no elements in it; otherwise, the DROP command will not be executed.

SYNTAX: DROP TABLE STUDENT CASCADE

2.2.1 Statements in SQL:

Following are the important statements used in SQL.

(i) SELECT - Used to retrieve the information from the relation.

(ii) INSERT - Used to insert the new values to the relation.

(iii) DELETE - used to delete one or more existing tuples from the relation.

(iv) UPDATE - Used to update already existing values in the relation.

2.2.2 Aggregate Functions in SQL:

Following aggregate functions are provided by the SQL.

(i) COUNT - Returns number of tuples.

(ii) SUM - Returns sum of entries in a column.

(iii) MAX - Returns Maximum value from an entire column.

(iv) MIN - Returns Minimum value from an entire column.

(v) AVG - Returns Average of all the entries in a column.

2.2.3 Constraints in SQL:

Following constraints are provided by the SQL.

(i) NOT NULL - Column should contain some value.

(ii) PRIMARY KEY - Should not allow duplicate and null values to a column.

(iii) UNIQUE - Each value of a column should be unique.

CHAPTER 3

FRONT END

The front end is an interface between the user and the back end. The front and back ends may be distributed amongst one or more systems. In network computing, *front end* can refer to any hardware that optimizes or protects network traffic. It is called application front-end hardware because it is placed on the network's outward-facing front end or boundary. Network traffic passes through the front-end hardware before entering the network. In compilers, the front end translates a computer programming source code into an intermediate representation, and the back end works with the intermediate representation to produce code in a computer output language.

The back end usually optimizes to produce code that runs faster. The front-end/back-end distinction can separate the parser section that deals with source code and the back end that optimizes. These days, front-end development refers to the part of the web users interact with. In the past, web development consisted of people who worked with Photoshop and those who could code HTML and CSS.

Now, developers need a handle of programs like Photoshop and be able to code not only in HTML and CSS, but also JavaScript or jQuery, which is a compiled library of JavaScript. Most of everything you see on any website is a mixture of HTML, CSS, and JavaScript, which are all controlled by the browser. For example, if you're using Google Chrome or Firefox, the browser is what translates all of the code in a manner for you to see and with which to interact, such as fonts, colours, drop-down menus, sliders, forms, etc. In order for all of this to work, though, there has to be something to support the front-end; this is where the backend comes into play

3.1 HTML:

HTML or **Hyper Text Mark-up Language** is the standard mark-up language used to create web pages.HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The

first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a mark-up language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behaviour of HTML web pages.

HTML is descriptive mark-up language. Library of various mark-up languages is defined in various browsers.

a) HTML Images - The <img> Tag and the Src Attribute:

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

Syntax for defining an image:

<imgsrc="*url*" alt="*some_text*">

b) HTML FORMS:

HTML forms are used to pass data to a server. An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, text area, field set, legend, and label elements.

The <form> tag is used to create an HTML form:

<form>
.
input elements
.
</form>

c) Image tag (<img>):

To add an image to an HTML document, we just need to include an <IMG> tag with a reference to the desired image. The <IMG> tag is an empty element i.e. it doesn't require a closing tag and we can use it to include from small icons to large images.

Syntax: <imgsrc="URL" alt=" alternative text">

d) HTML Lists:

An ordered list:

- The first list item
- The second list item
- The third list item

An unordered list:

- List item
- List item
- List item

3.1.1 HTML 5:

HTML5 will be the new standard for HTML. The previous version of HTML, HTML 4.01, came in 1999. The web has changed a lot since then. HTML5 is still a work in progress. However, the major browsers support many of the new HTML5 elements and APIs.

HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

Some rules for HTML5 were established:

a) New features should be based on HTML, CSS, DOM, and JavaScript

b) Reduce the need for external plug-ins (like Flash)

c) Better error handling

d) More mark-up to replace scripting

e) HTML5 should be device independent

f) The development process should be visible to the public

3.2 CSS:

CSS tutorial or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a mark-up language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages' use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

With plain HTML you define the colours and sizes of text and tables throughout your pages. If you want to change a certain element you will therefore have to work your way through the document and change it. With CSS you define the colours and sizes in "styles". Then as you write your documents you refer to the styles. Therefore: if you change a certain style it will change the look of your entire site. Another big advantage is that CSS offers much more detailed attributes than plain HTML for defining the look and feel of your site.

3.3 Javascript:

JavaScript (**JS**) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

JavaScript was formalized in the ECMA Script language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to objects within a host environment.

JavaScript is the most popular programming language in the world. It is the language for HTML, for the Web, for computers, servers, laptops, tablets, smart phones, and more. You can use JavaScript to:

a) Change HTML elements

- Delete HTML elements
- Create new HTML elements and to Copy and clone HTML elements.

CHAPTER 4

BACKEND

In a previous Chapter, we talked about how web programmers are concerned with launching websites, updates, and maintenance, among other things. All of that works to support the front-end of the website. The back-end has three parts to it: server, application, and database.

To better explain how all of this works, let's use the example of a customer trying to purchase a plane ticket using a website. Everything that the customer sees on the webpage is the front-end, as we have explained before, but once that customer enters all of his or her information, such as their name, billing address, destination, etc., the web application stores the information in a database that was created previously on the server in which the website is calling for information.

The web application creates, deletes, changes, renames, etc items in the database. For example, when a customer purchases a ticket, that creates an item in the database, but when they have a change in their order or they wish to cancel, the item in the database is changed.

In short, when a customer wants to buy a ticket, the backend operation is the web application communicating with the server to make a change in a database stored on said server. Technologies like PHP, Ruby, Python, and others are the ones backend programmers use to make this communication work smoothly, allowing the customer to purchase his or her ticket with ease.

4.1 MySQL's Logical Architecture:

The topmost layer contains the services that aren't unique to MySQL. They're services most network-based client/server tools or servers need: connection handling, authentication, security, and so forth.

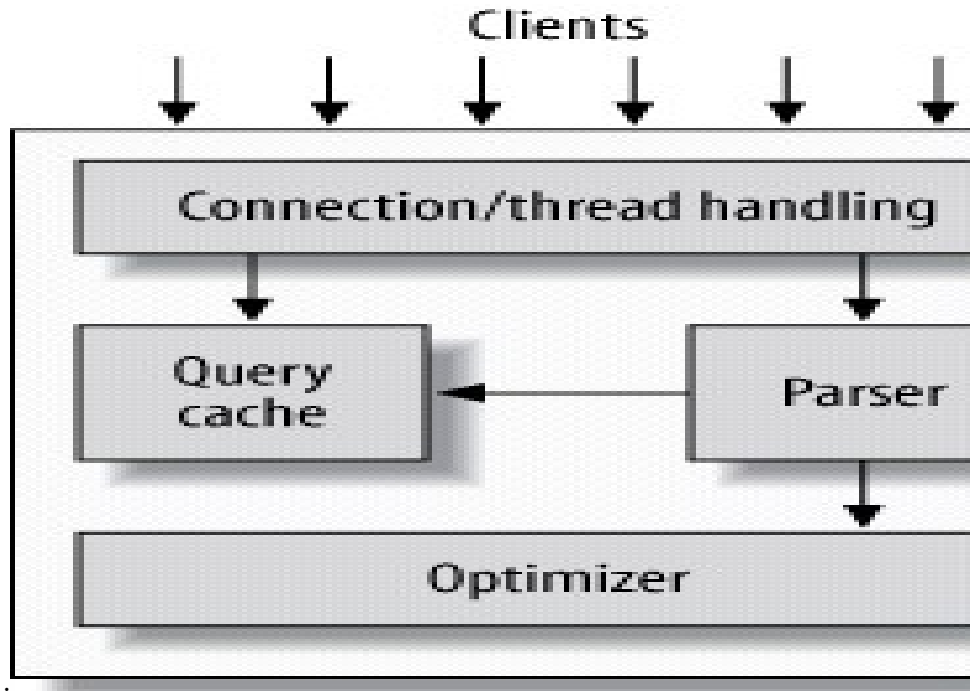


Fig. 4.1 MySQL's Logical Architecture

The third layer contains the storage engines. They are responsible for storing and retrieving all data stored "in" MySQL. Like the various file systems available for GNU/Linux, each storage engine has its own benefits and drawbacks. The server communicates with them through the storage engine API. This interface hides differences between storage engines and makes them largely transparent at the query layer. The API contains a couple of dozen low-level functions that perform operations such as begin a transaction" or "fetch the row that has this primary key.The storage engines don't parse SQL or communicate with each other; they simply respond to requests from the server.

4.2 Node js Architecture:

Node JS applications uses "Single Threaded Event Loop Model" architecture to handle multiple concurrent clients. There are many web application technologies like JSP, Spring MVC, ASP.NET, HTML, Ajax, jQuery etc. But all these technologies follow "Multi-Threaded Request-Response" architecture to handle multiple concurrent clients.

Any Web Application developed without Node JS, typically follows "Multi-Threaded Request-Response" model. Simply we can call this model as Request/Response Model. Client sends request to the server, then server do some processing based on clients request, prepare response and send it back to the client. This model uses HTTP protocol. As HTTP is a Stateless Protocol, this Request/Response model is also Stateless Model. So we can call this as Request/Response Stateless Model.

4.2.1 Drawbacks of Request/Response Stateless Model:

- Handling more and more concurrent client's request is bit tough.
- When Concurrent client requests increases, then it should use more and more threads, finally they eat up more memory.
- Sometimes, Client's Request should wait for available threads to process their requests.
- Wastes time in processing Blocking IO Tasks.

Node JS Platform does not follow Request/Response Multi-Threaded Stateless Model. It follows Single Threaded with Event Loop Model. Node JS Processing model mainly based on Javascript Event based model with Javascript callback mechanism. As Node JS follows this architecture, it can handle more and more concurrent client requests very easily.

4.2.2 Single Threaded Event Loop Model Processing Steps:

- Clients Send request to Web Server.
- Node JS Web Server internally maintains a Limited Thread pool to provide services to the Client Requests.
- Node JS Web Server receives those requests and places them into a Queue. It is known as "Event Queue".
- Node JS Web Server internally has a Component, known as "Event Loop". Why it got this name is that it uses indefinite loop to receive requests and process them. (See some Java Pseudo code to understand this below).
- Event Loop uses Single Thread only. It is main heart of Node JS Platform Processing Model.
- Even Loop checks any Client Request is placed in Event Queue. If no, then wait for incoming requests for indefinitely.
- If yes, then pick up one Client Request from Event Queue

- Starts process that Client Request
- If that Client Request Does Not requires any Blocking IO Operations, then process everything, prepare response and send it back to client.
- If that Client Request requires some Blocking IO Operations like interacting with Database, File System, External Services then it will follow different approach
    - Checks Threads availability from Internal Thread Pool
    - Picks up one Thread and assign this Client Request to that thread.
    - That Thread is responsible for taking that request, process it, perform Blocking IO operations, prepare response and send it back to the Event Loop
    - Event Loop in turn, sends that Response to the respective Client.

4.2.3 Single Threaded Event Loop Advantages:

- Handling more and more concurrent client's request is very easy.
- Even though our Node JS Application receives more and more Concurrent client requests, there is no need of creating more and more threads, because of Event loop.
- Node JS application uses less Threads so that it can utilize only less resources or memory

4.3 Express:

Express is a fast, assertive, essential and moderate web framework of Node.js. You can assume express as a layer built on the top of the Node.js that helps manage a server and routes. It provides a robust set of features to develop web and mobile applications.

The core features of Express framework:

- It can be used to design single-page, multi-page and hybrid web applications.
- It allows to setup middlewares to respond to HTTP Requests.
- It defines a routing table which is used to perform different actions based on HTTP method and URL.
- It allows to dynamically render HTML Pages based on passing arguments to templates.

4.3.1 Advantages of Express:

- Ultra fast I/O
- Asynchronous and single threaded
- MVC like structure
- Robust API makes routing easy

CHAPTER 5

SYSTEM REQUIREMENTS

This chapter will discuss the system requirements to develop this project.

5.1 Hardware Requirements

- System: Pentium i5 2.4 GHz.
- Hard Disk: 1TB.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- Ram: 2GB.

5.2 Software Requirements

- Operating system: Windows 10.
- Front end: HTML, CSS, Javascript.
- Back end: MySQL, Node js, Express.

CHAPTER 6

SYSTEM DESIGN

This chapter will discuss the system design used in this project.

6.1 ER diagram:

An Entity-Relationship (ER) model is an abstract way to describe a database. It is a popular high-level conceptual data model. Entity relationship diagrams (ER diagrams) are used to present the diagrammatic notations associated with ER model.

6.1.1 Notations for ER diagram:

The notations used for the ER Diagram for the project is as follows:

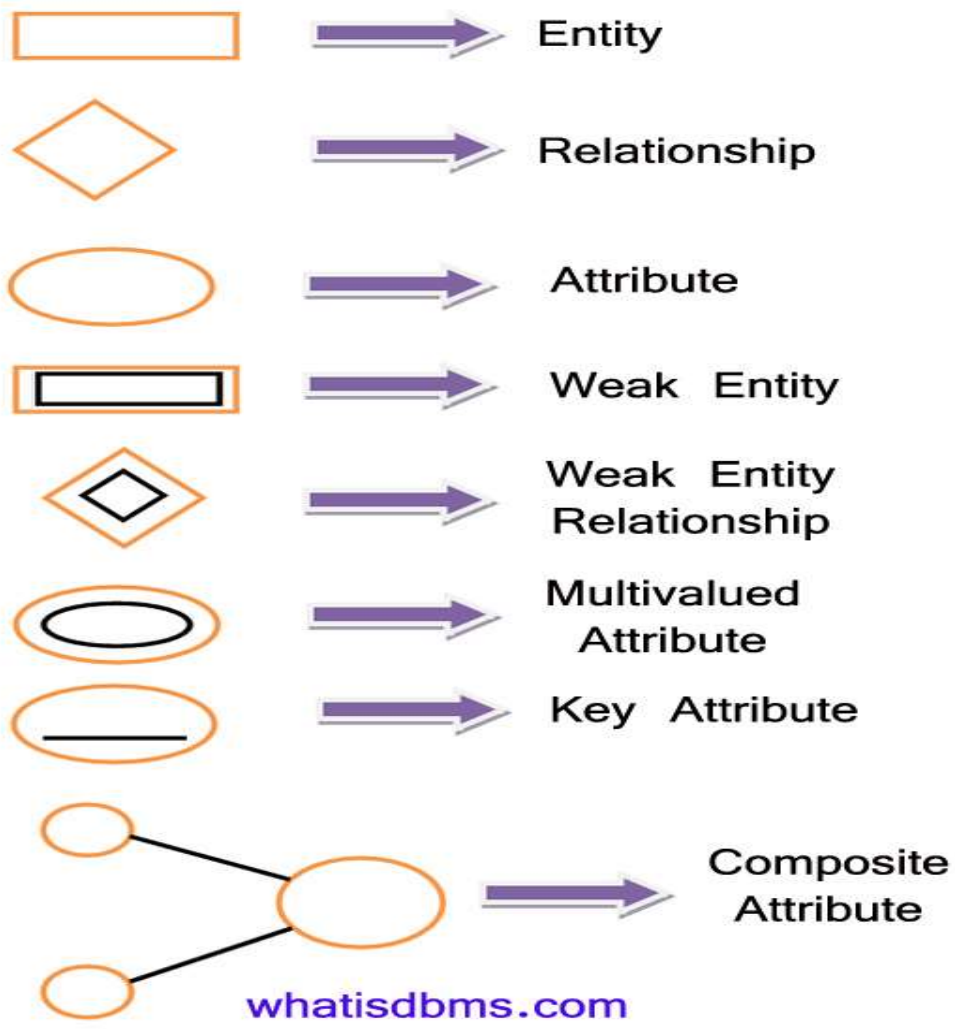


Fig.6.1.1 Notations for ER Diagram

6.1.2 ER Diagram for Zayka:

The ER Diagram for the project is as follows:

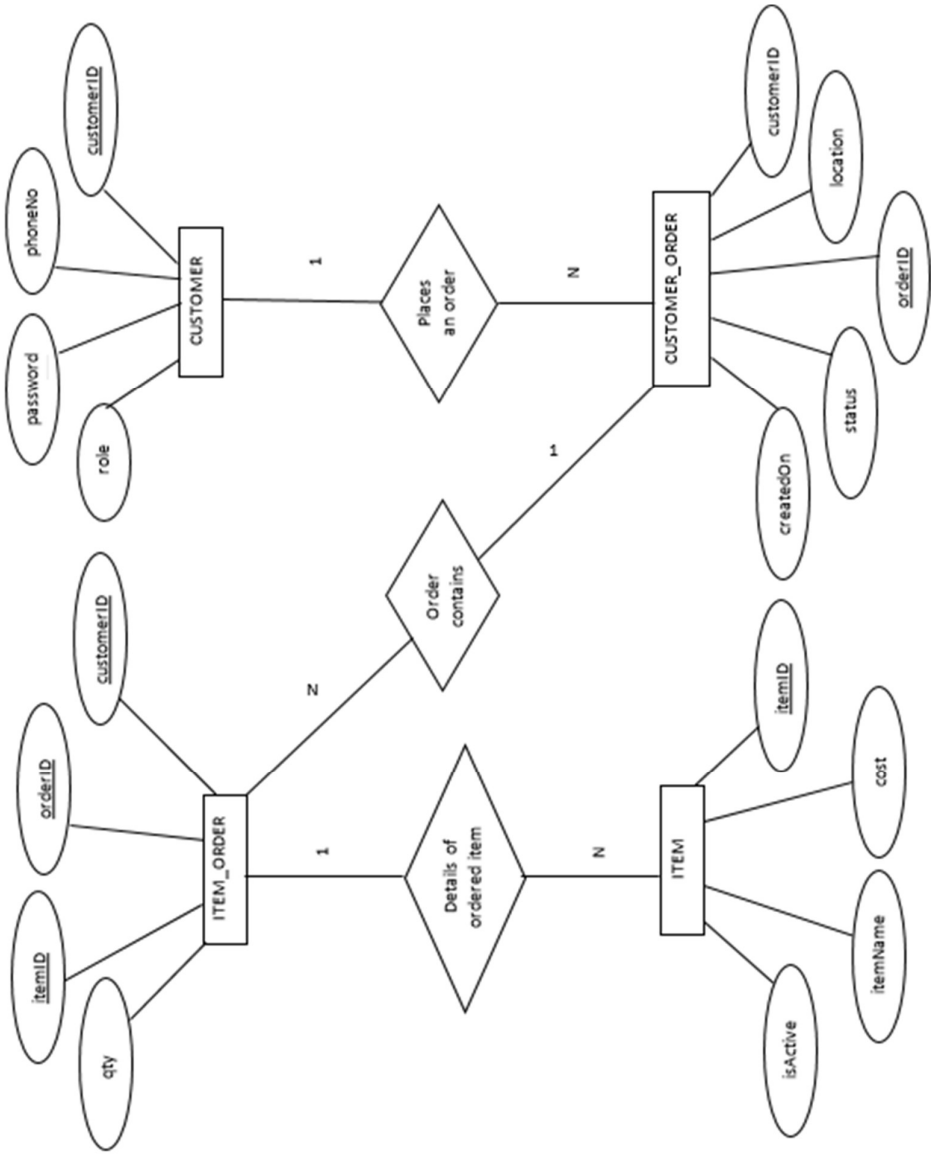


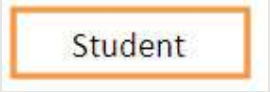
Fig.6.1.2 ER Diagram for Zayka

6.1.3 Components of an E-R diagram:

An E-R diagram constitutes of following Components

A. Entity: - Any real-world object can be represented as an entity about which data can be stored in a database. All the real world objects like a book, an organization, a product, a car, a

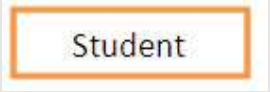
person are the examples of an entity. Any living or non-living objects can be represented by an entity. An entity is symbolically represented by a rectangle enclosing its name.



Entity

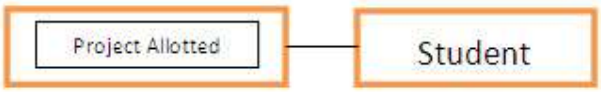
Entities can be characterized into two types:

- **Strong entity***:* A strong entity has a primary key attribute which uniquely identifies each entity. Symbol of strong entity is same as an entity.



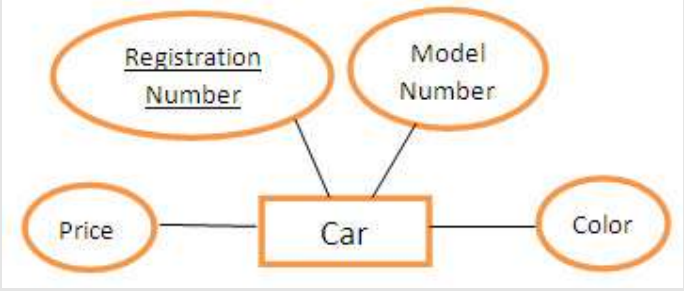
Strong Entity

- **Weak entity**: A weak entity does not have a primary key attribute and depends on other entity via a foreign key attribute.



Weak Entity

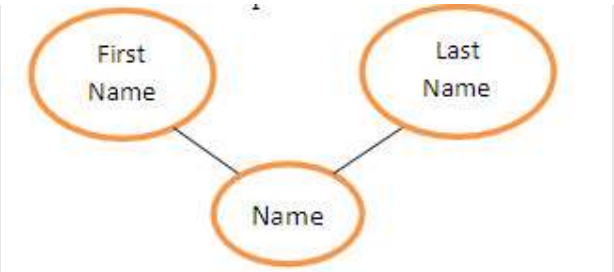
B. Attribute: - Each entity has a set of properties. These properties of each entity are termed as attributes. For example, a car entity would be described by attributes such as price, registration number, model number, colour etc. Attributes are indicated by ovals in an e-r diagram.



Attribute

A primary key attribute is depicted by an underline in the e-r diagram. An attribute can be characterized into following types:

- **Simple attribute**: - An attribute is classified as a simple attribute if it cannot be partitioned into smaller components. For example, age and sex of a person. A simple attribute is represented by an oval.
- **Composite attribute**: - A composite attribute can be subdivided into smaller components which further form attributes. For example, 'name' attribute of an entity "person" can be broken down into first name and last name which further form attributes. Grouping of these related attributes forms a composite attribute. 'name is the composite attribute in this example.



Composite Attribute

- **Single valued attribute**: - If an attribute of a particular entity represents single value for each instance, then it is called a single-valued attribute. For example, Ramesh, Kamal and Suraj are the instances of entity 'student' and each of them is issued a separate roll number. A single oval is used to represent this attribute.
- **Multi valued attribute*:*** – An attribute which can hold more than one value, it is then termed as multi-valued attribute. For example, phone number of a person. Symbol of multi-valued attribute is shown below,

Multi Valued Attribute

- **Derived attribute*:*** A derived attribute calculates its value from another attribute. For example, 'age' is a derived attribute if it calculates its value from 'current date' & 'birth date' attributes. A derived attribute is represented by a dashed oval.

C. Relationships: - A relationship is defined as bond or attachment between 2 or more entities. Normally, a verb in a sentence signifies a relationship.

For example,

- An employee assigned a project.
- Teacher teaches a student.
- Author writes a book.

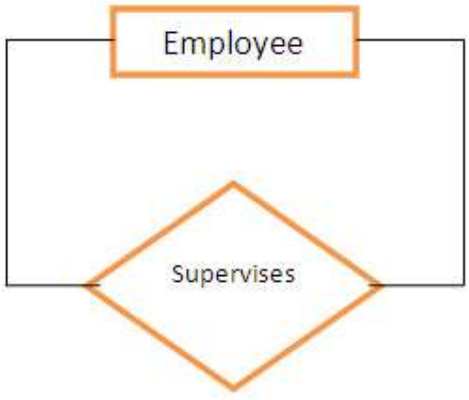
A diamond is used to symbolically represent a relationship in the e-r diagram.

Relationship

Various terms related to relationships

a) Degree of relationship: - It signifies the number of entities involved in a relationship. Degree of a relationship can be classified into following types:

- **Unary relationship**: - If only single entity is involved in a relationship then it is a unary relationship. For example, An employee (manager) supervises another employee.



Unary relationship

- **Binary relationships**: - when two entities are associated to form a relation, then it is known as a binary relationship. For example, A person works in a company. Most of the times we use only binary relationship in an e-r diagram. The teacher-student example shown above signifies a binary relationship.

Other types of relationships are ternary and quaternary. As the name signifies, a ternary relationship is associated with three entities and a quaternary relationship is associated with four entities.

b) Connectivity of a relationship: - Connectivity of a relationship describes, how many instances of one entity type are linked to how many instances of another entity type. Various categories of connectivity of a relationship are;

- **One to One (1:1)** – "Student allotted a project" signifies a one-to-one relationship because only one instance of an entity is related with exactly one instance of another entity type.

One To One

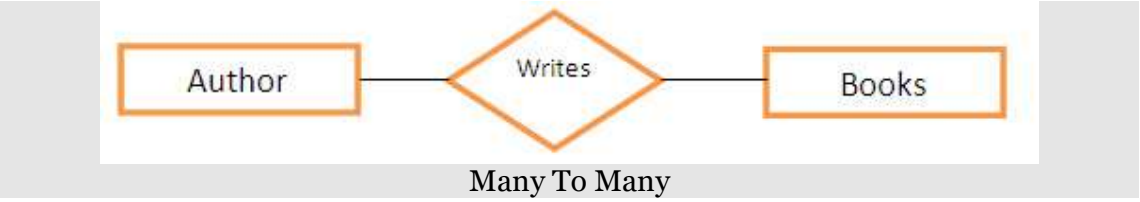
- **One to Many (1:M)** – "A department recruit's faculty" is a one-to-many relationship because a department can recruit more than one faculty, but a faculty member is related to only one department.

One to Many

- **Many to One (M:1)** – "Many houses are owned by a person" is a many-to-one relationship because a person can own many houses but a particular house is owned only a person.

Many To One

- **Many to Many (M: N)** – "Author writes books" is a many-to-many relationship because an author can write many books and a book can be written by many authors.



Many To Many

6.2 Tables Implemented:

The tables implemented in this application are as follows:

CUSTOMER

NAME	TYPE	NULL	KEY	DEFAULT
customerID	int(11)	NOT NULL	PRIMARY KEY	
phoneNo	varchar(10)	NOT NULL		
password	varchar(50)	NOT NULL		
role	int(1)	NOT NULL		0

CUSTOMER_ORDER

NAME	TYPE	NULL	KEY	DEFAULT
orderID	int(11)	NOT NULL	PRIMARY KEY	
customerID	int(11)	NOT NULL		
location	varchar(50)	NULL		
createdOn	datetime	NULL		CURRENT_TIME_

				STAMP
status	int(11)	NOT NULL		0

ITEM

NAME	TYPE	NULL	KEY	DEFAULT
itemID	int(11)	NOT NULL	PRIMARY KEY	
itemName	varchar(50)	NOT NULL		
cost	double	NOT NULL		
isActive	int(1)	NOT NULL		1

ITEM_ORDER

NAME	TYPE	NULL	KEY	DEFAULT
orderID	varchar(15)	NOT NULL	PRIMARY KEY	
customerID	int	NOT NULL	PRIMARY KEY	
itemID	date	NOT NULL	PRIMARY KEY	
qty	int(11)	NOT NULL		0

6.3 Schema Diagram:

The description of database is called the Database Schema, which is specified during database design and is not expected to change frequently. Most data models have certain conventions for displaying schemas as diagrams. A displayed schema is called a schema diagram.

A **database schema** of a database system is its structure described in a formal language supported by the database management system (DBMS) and refers to the organization of data to create a blueprint of how a database will be constructed (divided

into database tables). The formal definition of database schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real world entities are modeled in the database.

6.3.1 Schema Diagram for Zayka:

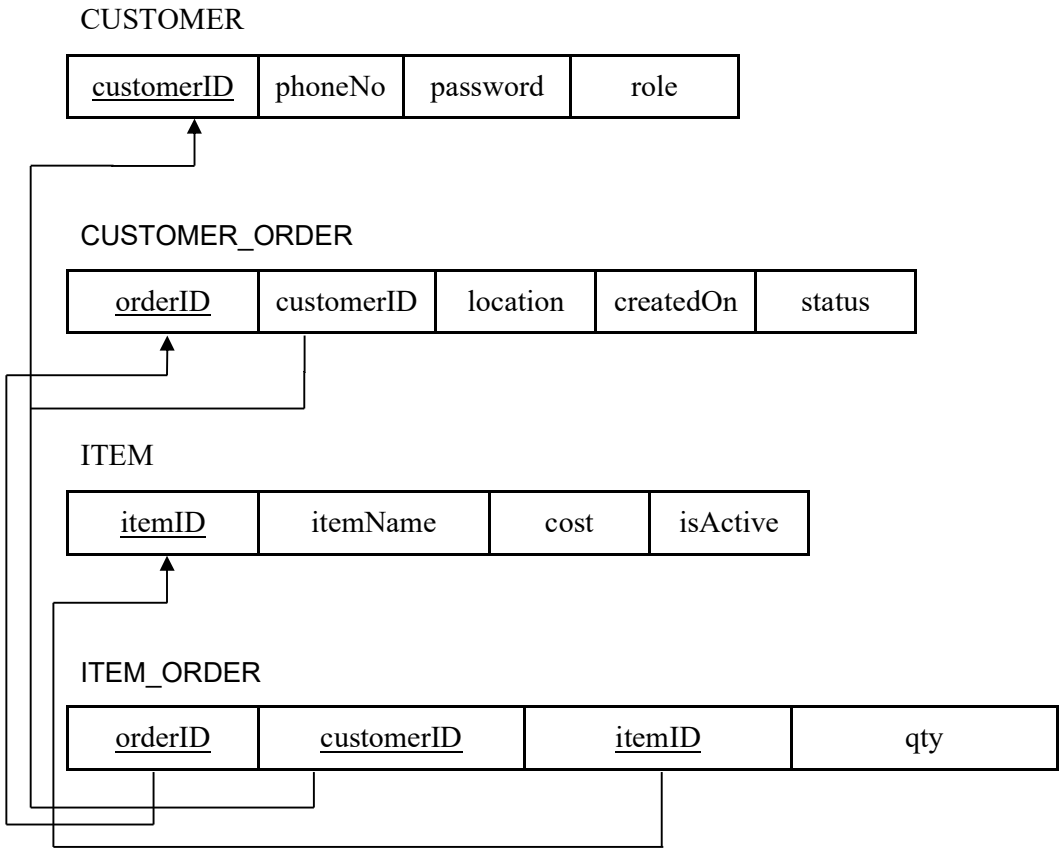


Fig.6.1.3 Schema Diagram for Zayka

CHAPTER 7

SOFTWARES AND TOOLS USED

This chapter will discuss the software's and tools used in this project.

7.1 Introduction to My SQL:

The database has become an integral part of almost every human's life. Without it, many things we do would become very tedious, perhaps impossible tasks. Banks, universities, and libraries are three examples of organizations that depend heavily on some sort of database system. On the Internet, search engines, online shopping, and even the website naming convention would be impossible without the use of a database. A database that is implemented and interfaced on a computer is often termed a database server.

One of the fastest SQL (Structured Query Language) database servers currently on the market is the MySQL server, developed by T.c.X. DataKonsultAB. MySQL, available for download at www.mysql.com, offers the database programmer with an array of options and capabilities rarely seen in other database servers. MySQL is free of charge for those wishing to use it for private and commercial use. Those wishing to develop applications specifically using MySQL should consult MySQL's licensing section, as there is charge for licensing the product.

These capabilities range across a number of topics, including the following:

a) Ability to handle an unlimited number of simultaneous users.

b) Capacity to handle 50,000,000+ records.

c) Very fast command execution, perhaps the fastest to be found on the market.

d)Easy and efficient user privilege system.

However, perhaps the most interesting characteristic of all is the fact that it's free. That's right, T.c.X offers MySQL as a free product to the general public.

7.1.1 Reasons to Use MySQL:

The following are the reasons to use MySQL for this project.

a) Scalability and Flexibility:

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavours of Linux, UNIX, and Windows being supported.

b) High Performance:

A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results.

C) High Availability:

Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

d) Robust Transactional Support:

MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa.

e) Web and Data Warehouse Strengths:

MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data inserts capability, and strong support for specialized web functions like fast full text searches.

f) Strong Data Protection:

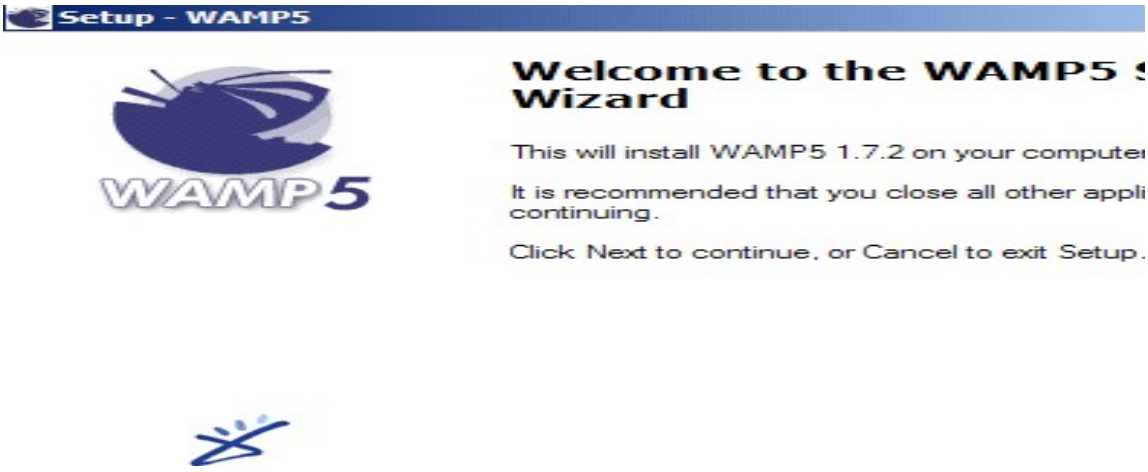
Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible.

g) Management Ease:

MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX.

7.2 WAMP:

a) Install WAMP by double clicking on the icon, an installation wizard will be opened.



b) Click on next button to continue, and then again box will be appeared on the screen asking you for acceptance or not acceptance of the license agreement.

c) For this select option “I accept the terms in the license agreement”. Click on next button.

d) Then again wizard will be opened asking you about the location of placing the folder. For selecting location click on browse and click next.

e) The next wizard will ask you about auto start the WAMP i.e. if you want to start the WAMP automatically then click on the checkbox and if do not want then it remain unchecked.

f) Click on next button to precede further, the next wizard will display you the summary of the setting. Click on install button for installation.

g) After installation the next wizard will be opened asking you about directory for your root folder. if you are not sure, just leave the default directory. for proceeding further click on next button.

h) The next wizard will ask you about your server just fill localhost and click on next button.

i) The next wizard will be appeared on the screen asking you about the email address for sending mails. Just leave the default email address if you are not sure. Click on the next button.

j) wizard will ask you about the browser by default browser is internet explorer you can set default browser according to your requirement. click on next button.

k) On clicking next a popup menu will have displayed asking you “would you like to install the new WAMP homepage?” click on yes option.

l) Finally click on finish after complete installation of WAMP on your system.

CHAPTER 8

SNAPSHOTS

Fig 8.1 Home Page

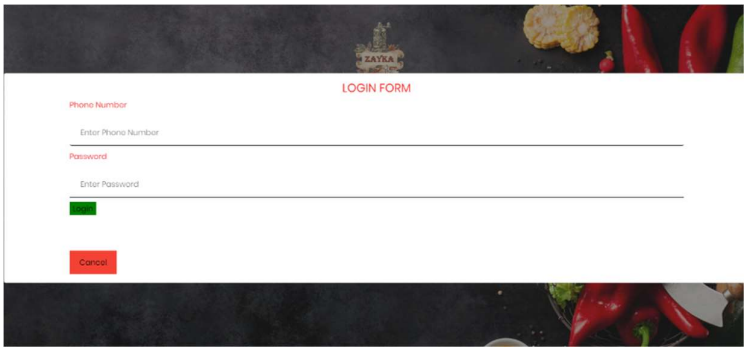


Fig 8.2 Login Form

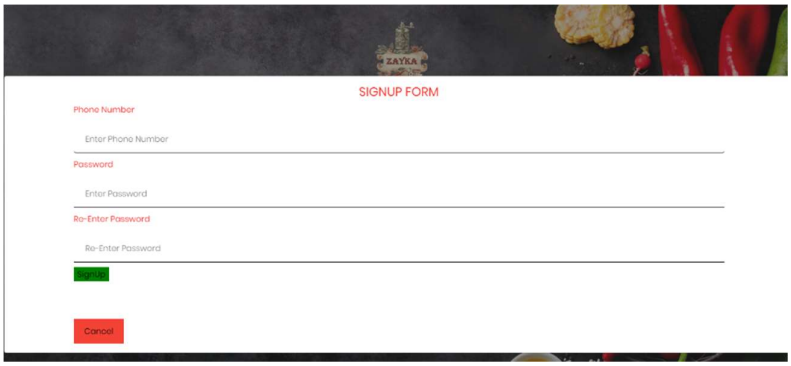


Fig 8.3 Signup Form

Fig. 8.4 Admin Page

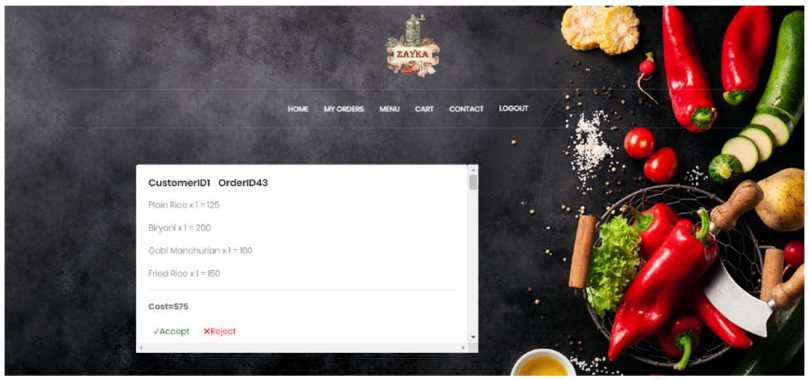


Fig. 8.5 Admin: Order Requests Page

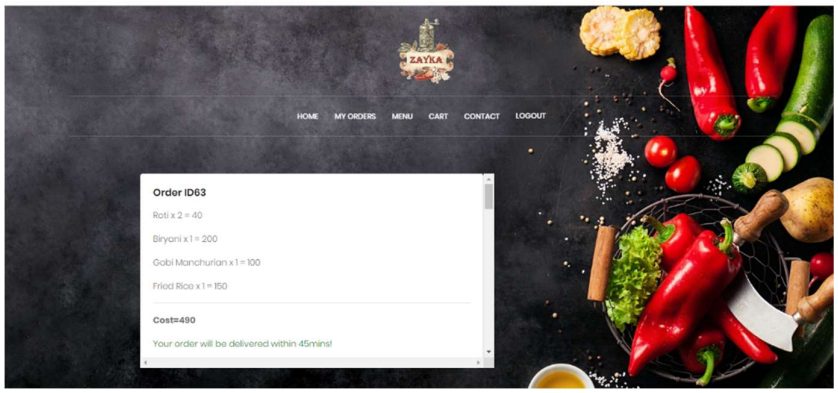


Fig. 8.6 Order Summary

CONCLUSIONS

This project is carried out to analyze the internet applications on database management through the use of MySQL and the features of NodeJS and the way it can be used and interfaced with MySQL as a standalone database application.

This project was made to make it easier for customers to order food online from their favorite restaurant. The authorization and authentication is kept minimalistic to make it easier and faster for the customers to order food online. The user can check the status of his current order and also can view his previous orders. The super admin can add one of his employee to be admin, which will enable him/her to add new items to the menu and also accept or reject customer orders.

The project Zayka is a restaurant specific and it meets the requirements of a user. This project has a scope for enhancements such as providing administration facilities, recommendations for users, order location tracking facility and reservations.

REFERENCE

[1] Elmasri and Navathe, Fundamentals of Database Systems, Fourth Edition, Addison-Wesley, 2005.

[2] www.journaldev.com

[3] http://www.w3schools.com

[4] www.npmjs.com

.