A

Practical Traning Project Report

On

**UDAIPUR EVENTZ**

*Submitted*

*in partial fulfilment*

*for the award of the Degree of*

**Bachelor of Technology**

in

**Computer Science & Engineering**

**with Specialization in Computer Science**



**Submitted To: Submitted By:**

**Dr. Prashant Sharma Student Name**

**Faculty of Engineering**

**Department of Computer Engineering**

**Pacific University, Udaipur , Rajsthan**

1

**CERTIFICATE**

This is certify that the Project entitled “**UDAIPUR EVENTZ**” is a bona-fide report of the work carried out by **Student Name (C.S.E.)** under the guidance and supervision of **Dr.Prashant Sharma** for the award of the degree of Bachelor of Technology at Pacific University, Udaipur (Rajsthan).

To the best of my knowledge and belief, this work embodies the work of candidate themselves, has duty been completed, fulfils the requirement of ordinance relating to the Bachelor Degree of the University and is up to the standard in respect of content, presented and language for being referred to the examiner.

**Dr. Prashant Sharma**

HOD(Department of Computer Science & Engineering)

Pacific Institute of Technology

Udaipur(Rajsthan)

**ABSTRACT**

Our project explains about the Book an Event Organiser in Udaipur out of many choices. This project mainly explains the various actions related to Event Organisers and Customers. This project shows some ease in adding, editing and deleting the Events Bussiness/Compnay. It also provides a less time consuming process for viewing, adding, editing and deleting the details of Event Organisers

Our project includes

* Event Organisers/Owner Registration
* Customer Get Best Results
* Admin Operations

**ABOUT THE COMPANY**

I have completed my Training from **Maharaja Infotech pvt. Ltd. , Ajmer.** under the guidance of **Mr.Abhinav Sharma .**

#### **Mr.Abhinav Sharma** Completed there B.Tech. in Computer Science in 2014. After this they worked in **Kirnani Technologies** at the post of **Jr. Web Developer** for 1 Year.

After this, He started his own business in Web Development in 2015. After this, he continued the work and progressed in this line. After that Official Company is registered in January 2017 with the name of **Maharaja Infotech pvt. Ltd.**

**Mr.Abhinav Sharma** Registered his company with **Single Proprietorship** and register his Firm & Domain . Domain Category of **Service Provider**. Today in 2019 he is a Professional Web Developer , **SEO Expert** , Software Development Specialist and Certified IT Trainer. The Turnover of his company is 5 to 6 lakh per year. Company give the taxes. **Mr.Abhinav Sharma** is Self Certified for Training.

Today in the company have 3 ofrole employees and 6 onrole employees. The Company is ISO Certified iso 9001:2015.

Company is certified in following fields:

1. Training & Coaching
2. Software Design & Development
3. IT Services

Company have a registered Certificate for **Training & Coaching**. The Company is worked on **Govt. Approved Projects**. Toady in 2019 Company is worked on Approximate 3000+ Projects including Web Applications , Software Development etc.

Company have some following live Projects are working:

RoomOnRent

TakeMySearch

KaleraAyurveda



**1. INTRODUCTION**

**1.1 Introduction to Project**

**UDAIPUR EVENTZ** is website which is helpful for users for finding an Event Management Company in Udaipur . In the current system all the activities are done manually. It is very time consuming and costly. **UDAIPUR EVENTZ** deals with the many events oragnisers from Udaipur.

There are mainly 3 modules in this software

* User module
* Owner Module
* Admin Module

In the Website we can register as a user and user has of two types, Owner and Customer. A Owner can register as user and can add edit and delete his profile, Upload his event business. The administrator can add edit and delete Owner & Business Details. All the users can see the all event oraganisers.

**1.2 Purpose of the Project**

The Purpose of this project is that the world is going to be digital trend. So, problem is that the Event Oraganiser cannot get business manually. Because Udaipur City is Big anybody a have no time to search any event oragniser here & there, because this is time consuming. Here we can provide this Online Platform to Search an Event Oragniser in Udaipur Easily.

**1.3 Problem in Existing System**

The Problem in Existing System is that Owner of any Event Organisers not know how to promote business digitally and get business leads.

The Same Problem is with user that is they don’t know how to use internet for make life simpler & Easier.

**1.4 Solution of these Problem**

The Solutions of these problem is that guide the people for they can use smartphone **smartly** and use of smartphone for make life easier.

If **UDAIPUR EVENTZ** can promote their business make aware to people for using their website to make things easier. After this people can active for using **UDAIPUR EVENTZ** website.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS**

**2.1 Purpose**

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

**2.2 Scope**

**Scope** is the work to be done, is The features and functions that characterize a product, service, or result. The scope defines the boundaries of a project, what features will be included and implemented within this scope, what is the delivery dates and milestones need to be delivered as well the required budget to deliver that scope. If the project is poorly controlled and governed, then the scope creep is expected.

## 2.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Configuration | It means customer getting call to Event Oragniser. |
| FAQ | Frequently Asked Questions |
| CRM | Customer Relationship Management |

**2.4 Technologies**

**Technologies** used in this project is that, we can used web development technologies HTML, CSS, Javascript, Bootstrap , SQL and php languages. For making this project we can also used Editor Softwares Notepad++ and Sublime Text Editor. In this we have used **xampp**  software for access an local host server and database.

**2.5 External Interface Requirements**

According to Richard Thayer (2002), "External interface requirements specify hardware, software, or database elements with which a system or component must interface...." This section provides information to ensure that the system will communicate properly with external components. If different portions of the product have different external interfaces, incorporate an instance of this section within the detailed requirements for each such portion.

**2.5.1 Hardware Interface**

Describe the characteristics of each interface between the software and hardware components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used.

**2.5.2 Software Interface**

Describe the connections between this product and other software components (identified by name and version), including databases, operating systems, tools, libraries, and integrated commercial components. State the purpose of the messages, data, and control items exchanged between the software components. Describe the services needed by external software components and the nature of the intercomponent communications. Identify data that will be shared across software components. If the data-sharing mechanism must be implemented in a specific way, such as a global data area, specify this as a constraint.

**3. GENERAL DESCIPTION**

**3.1 Product Perspective**

This software product is eventually intended for the software developers. Product will be deployed to web site and all users of the product will access by use of the website. Website will be main user interface where users can operate all the provided functionality. However, this web site will be only a part of a larger system. There will be cloud server where all the user data is kept and all the execution is done. Website will only be the interface for the user data and the execution of provided functionalities. To use product, users are required to register through the web interface. Whenever a new user registered, all the required data will be created in the database and a predefined workspace will be assigned for the user. Later, user will be able to login and logout the system anytime he wants. Since every operation that user perform reflected to our database, user will find his workspace however he leaves last time. From the user point of view, user will have to functionality to create and edit files in his own workspace. User will be able to run predefined programs on these files such as language specific compilers and debuggers (gdb, g++, javac, etc) and project management tools (mvn, svn, git, etc). All of the files users created will be kept in cloud server and all the CPU requiring work such as compiling and running the programs will be executed on cloud server so that user will be able to access his own integrated development environment with his specific setting anywhere he wants.

**3.2 Product Functions**

This new product, web based integrated development environment, must have number of features which will allow users to use functionalities which have been explained above. Required functionalities of the product can be summarized in five categories; user management requirements, code editor requirements, debugger requirements, command line interface requirements and interface requirements. Overall description of the requirements can be found below;

**3.2.1 User Management Requirements**

This category of requirements is related to user authentication mechanism and workspace management of users. Each user will have credentials to connect their workspace on cloud and will be assigned to workspace. Users will perform all the functionality over this workspace using his credentials.

**3.2.2 Code Editor Requirements**

One of the most important functionality expected from an integrated development environment is a code editor which will ease the developer’s life. Code editor will be the main interface that developers deal with. It supports variety of programming language with highlighting, syntax checking, auto-indentation and language specific auto-complete.

**3.2.3 Debugger Requirements**

Debugger is the main tool that developers can test and debug their target program. Debugger of the product should allow setting and displaying breakpoints on the code. It will also provide functionality of stopping/continuing of the execution of debugger. Finally, it will provide an expression interface where user can enter an expression and observe the value of expression at each step.

**3.2.4 Terminal Requirements**

As an important part of the software development process, an integrated development environment should provide a command line interface where user can work in old fashion and accomplish complicated tasks such as configuring git synchronization. Main component of CLI will be the terminal. Terminal will allow user to run UNIX command on his own workspace and also run predefined programs such as mvn, svn etc. Terminal will also provide auto-complete by list of available commands and browse in the command history.

**3.2.5 Interface Requirements**

This group of requirements is related to external interaction of the workspace with outer world. For user to interact with the workspace, product will provide both command line interface and graphical interface. Command line interface will be UNIX like and graphical interface will allow tabbed navigation of windows, hierarchical view of workspace etc. Again as an external interface, product will support a synchronization interface for external services.

**3.3. User Characteristics**

Users of this web based integrated development environment will mainly be software developers. Since it is reasonable to assume that an average developer has knowledge about functionalities and usage of IDE, we assume that our users will already be informed about basic functionality of the product. Also clear documentation and tutorials about the product feature will be provided.

**3.4. Constraints**

Developers of the product should be aware that main feature of the intended product is portability. So they should use common libraries and tools that can work with all the common internet browser application with no problem. Developers should also be careful about the privacy of users. Since product will be cloud application, all user data will be kept on cloud server and necessary precautions should be taken to protect user data. Since product will be cloud application and all user programs will be executed on cloud server, developers should limit the privileges of the users so that they cannot harm other users’ data and system server.

**3.5 Performance Requirements**

Since this software is going to web – based, it does require a powerful server machine with high band internet access. Server machine should have a powerful CPU and high speed internet access so that it can handle multiple users at the same time. Another performance requirement is the storage space. Higher storage space means more user and bigger workspace per user so higher the storage, better the performance. Performance requirement by the user side is, web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections. Expected number of simultaneous user should be at least 100. System should be able to deal with 100 users at the same time. Also database of the system should handle at least a thousand of users at any periods.

**3.6 Specific Requirements**

we will describe the requirements of the software in detail. Basically, we will categorize requirements in 3 which are namely Performance Requirements requirements, security requirements and Portability requirements.

**3.6.1 Performance Requirements**

Since this software is going to web – based, it does require a powerful server machine with high band internet access. Server machine should have a powerful CPU and high speed internet access so that it can handle multiple users at the same time. Another performance requirement is the storage space. Higher storage space means more user and bigger workspace per user so higher the storage, better the performance. Performance requirement by the user side is, web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections. Expected number of simultaneous user should be at least 100. System should be able to deal with 100 users at the same time. Also database of the system should handle at least a thousand of users at any periods.

**3.6.2 Security Requirements**

Since this software will be hosted on cloud server, all the user data will be kept on the cloud server. Product should be able to protect privacy of user data. Workspace of the user should only be accessed through user own credentials and any other user should not be able to access to the user private data. Since execution will also be done in the machine in the cloud, user should be restricted in terms of user rights. User should only access to his own workspace and should not access to any other workspace with the programs they run on the cloud. Also rights of the user should be restricted so that user can not harm to system by the programs they run or by the commands they run on terminal. Since all the data will be transferred on the web, system should also use an encryption and decryption mechanism only intended user can decode the data and work on the data.

**3.6.3 Portability Requirements**

Main purpose of developing web-based IDE is to improve the portability of software development process. To improve portability, software should run on variety of platforms and variety of connection speeds. As explained in the performance requirements section, software should be lightweight so that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase. Such as using JavaScript and HTML5 instead of Apache Flex. 30 Portability also means running on most number of different platform without an additional effort. To achieve this, web application should be developed by using the common technologies and tools which are provided by all common web browsers and operating system such as HTML5, Javascript etc.

**4. SELECTED SOFTWARE**

**4.1 SQL Server**

SQL Server is Microsoft's relational database management system (RDBMS). It is a full-featured database primarily designed to compete against competitors Oracle Database (DB) and MySQL.

Like all major RBDMS, SQL Server supports ANSI SQL, the standard SQL language. However, SQL Server also contains T-SQL, its own SQL implemention. SQL Server Management Studio (SSMS) (previously known as Enterprise Manager) is SQL Server's main interface tool, and it supports 32-bit and 64-bit environments.

SQL Server is sometimes referred to as MSSQL and Microsoft SQL Server.

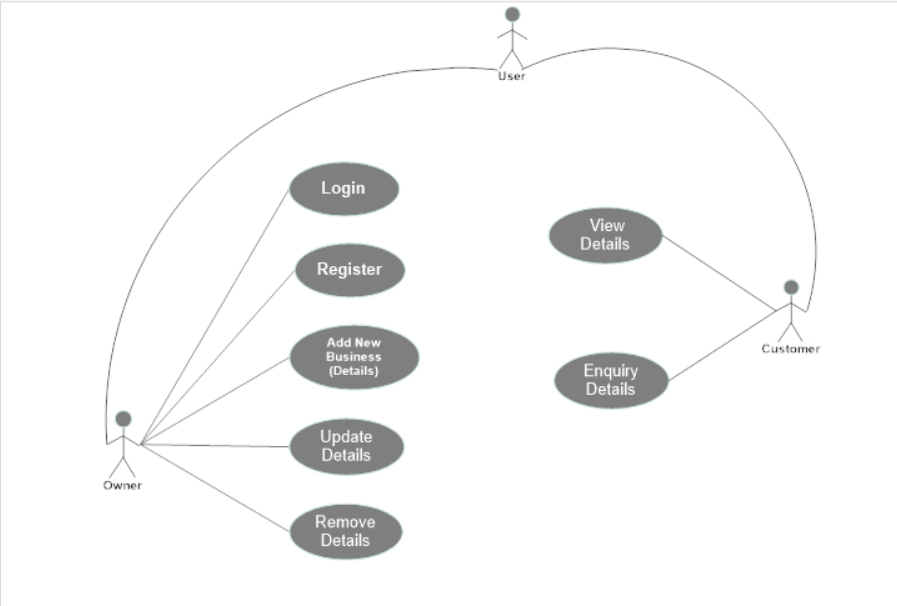
Originally released in 1989 as version 1.0 by Microsoft, in conjunction with Sybase, SQL Server and its early versions were very similar to Sybase. However, the Microsoft-Sybase partnership dissolved in the early 1990s, and Microsoft retained the rights to the SQL Server trade name. Since then, Microsoft has released 2000, 2005 and 2008 versions, which feature more advanced options and better security.

Examples of some features include: XML data type support, dynamic management views (DMVs), full-text search capability and database mirroring.

SQL Server is offered in several editions with different feature set and pricing options to meet a variety of user needs, including the following:

* Enterprise: Designed for large enterprises with complex data requirements, data warehousing and Web-enabled databases. Has all the features of SQL Server, and its license pricing is the most expensive.
* Standard: Targeted toward small and medium organizations. Also supports e-commerce and data warehousing.
* Workgroup: For small organizations. No size or user limits and may be used as the backend database for small Web servers or branch offices.
* Express: Free for distribution. Has the fewest number of features and limits database size and users. May be used as a replacement for an Access database.

**remain5. ACTORS AND USE CASE**

****

**6. WHAT IS WEBSITE ?**

**6.1 Website**

* Website is a collection of related web pages that may contain text images, audio and video. A Website can consist of one page, or thousands of pages, depending on what the site owner is trying to accomplish.
* Websites come in a nearly endless variety, including educational sites, news sites, forums, social media sites, e-commerce sites, and so on. The pages within a website are usually a mix of text and other media. That said, there are no rules dictating the form of a website. A person could create a website of nothing but black and white photos of roses, or the word "cat" linked to another Web page with the word "mouse." However, many sites follow a standard pattern of a homepage that links off to other categories and content within the website. Originally, websites were categorized by their top-level domains. Some examples include: Government agency websites = .gov Educational institutions’ websites = .edu Nonprofit organizations’ websites = .org Commercial websites = .com nformation sites = .info Although these top-level domains extensions still exist, they say little about a website's actual content. In the modern day internet, the ".com" extension is by far the most popular domain, a long with many other country-specific extensions.

## When was the first website created?

The first website was built at [**CERN**](https://www.computerhope.com/jargon/c/cern.htm)by [**Tim Berners-Lee**](https://www.computerhope.com/people/tim_berners-lee.htm) and launched on August 6, [1991](https://www.computerhope.com/history/1991.htm).

## How many websites are on the Internet?

As of **January 2018**, depending on which survey or hosting company being referenced, there are between **1.3 and 1.8 billion** websites on the Internet. Many of these websites are unused or not visited by many people, but the websites still exist and included in the count.

## What is the difference between a website and a web page?

A website refers to a central location that contains more than one web page or a series of web pages. For example **RoomOnRent** is considered a website, which contains thousands of different web pages.

[https://**www.**roomonrent.net/**tenant-profile**/](https://www.roomonrent.net/tenant-profile/)

Subdomain Directories /Web pages

Protocol Domain & Domain Suffix

## Who creates websites on the Internet?

Any business, government, organization, or person can create a website on the Internet. Today, the Internet consists of billions of websites created by billions of different people. You can even create a website or [blog](https://www.computerhope.com/jargon/w/weblog.htm) on the Internet.

**7. LANGUAGES USED**

In My Project “**UDAIPUR EVENTZ**” website have 6 Web Development Scripting Languages used. They are following:-

7.1. **HTML**(HyperText Markup Language)

7.2. **CSS**( Cascading Style Sheets )

7.3. **JAVASCRIPT**

7.4. **BOOTSTRAP**

7.5. **SQL**(Structured Query Language)

7.6. **PHP**(Hypertext Preprocessor)

**7.1. HTML:** HTML is the standard markup language for creating Web pages.

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
* Browsers do not display the HTML tags, but use them to render the content of the page

**Structure of HTML**

# <!DOCTYPE html> My First Heading

<html> My first paragraph.  
<head>  
<title>Page Title</title>  
</head>  
<body>  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
  
</body>  
</html>

### **Example Explained**

* The **<!DOCTYPE html>** declaration defines this document to be HTML5
* The **<html>** element is the root element of an HTML page
* The **<head>** element contains meta information about the document
* The **<title>** element specifies a title for the document
* The **<body>** element contains the visible page content
* The **<h1>** element defines a large heading
* The **<p>** element defines a paragraph

### **HTML Tags**

HTML tags are element names surrounded by angle brackets:

**<tagname>** content goes here...**</tagname>**

* HTML tags normally come in pairs like **<p>** and **</p>**
* The first tag in a pair is the **start tag**, the second tag is the **end tag**
* The end tag is written like the start tag, but with a **forward slash** inserted before the tag name
* The start tag is also called the **opening tag**, and the end tag the **closing tag**.

### **The <!DOCTYPE> Declaration**

* The **<!DOCTYPE>** declaration represents the document type, and helps browsers to display web pages correctly.
* It must only appear once, at the top of the page (before any HTML tags).
* The **<!DOCTYPE>** declaration is not case sensitive.
* The **<!DOCTYPE>** declaration for HTML5 is:

<!DOCTYPE html>

## Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document:



**HTML Versions**

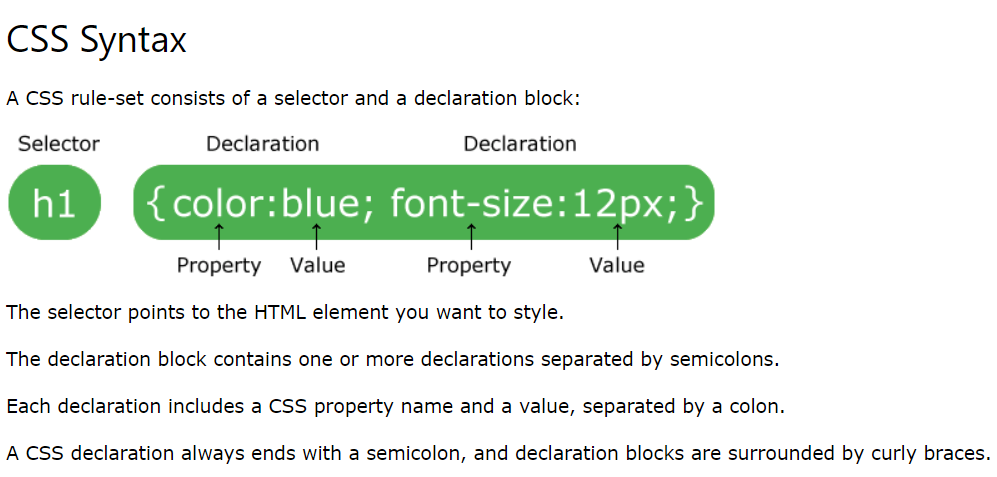
Since the early days of the web, there have been many versions of HTML:

|  |  |
| --- | --- |
| **Version** | **Year** |
| HTML | 1991 |
| HTML 2.0 | 1995 |
| HTML 3.2 | 1997 |
| HTML 4.01 | 1999 |
| XHTML | 2000 |
| HTML5 | 2014 |

**7.2 CSS:** CSS stands for Cascading Style Sheets

* It describes how HTML elements are to be displayed on screen, paper, or in other media
* It saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in **CSS** files
* **CSS** is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.
* The style definitions are normally saved in external **.css** files.
* With an external stylesheet file, we can change the look of an entire website by changing just one file!

**Structure of CSS**



**CSS Comments**

* Comments are used to explain the code, and may help when you edit the source code at a later date.
* Comments are ignored by browsers.

### **Example**

A CSS comment starts with **/\* and ends with \*/.** Comments can also span multiple lines:

p {  
  color: red;  
  **/\* This is a single-line comment \*/**  text-align: center;  
}  
  
**/\* This is  
a multi-line  
comment \*/**

**Example of CSS**

<!DOCTYPE html>

<html>  
 <head>  
 <style>

**p {**

**color: red;**

**text-align: center;**

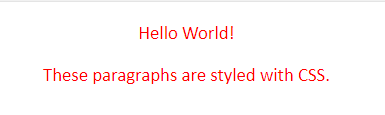
**}**

</style>  
 </head>  
<body>

/\*-----CSS Comments----\*/  
 <p>Hello World!</p>

<p>These paragraphs are styled with CSS.</p>   
</body>  
</html>

**OUTPUT:**



**CSS Selectors**

## The CSS id Selector

* The id selector uses the id attribute of an HTML element to select a specific element.
* The id of an element is unique within a page, so the id selector is used to select one unique element!
* To select an element with a specific id, write a hash (#) character, followed by the id of the element.

## The CSS class Selector

* The class selector selects HTML elements with a specific class attribute.
* To select elements with a specific class, write a period (.) character, followed by the class name.

**Example**

<html>

<style>

**#top {**

**color: red;**

**}**

**#top1 {**

**font-size:70px;**

**}**

**.top {**

**color: red;**

**}**

**.top1 {**

**font-size:70px;**

**}**

</style>

<h1 id="top top1">Chocolate curry</h1>

<h1 class="top top1">Chocolate curry</h1>

</html>

**Output:**

**Chocolate curry**

**Chocolate curry**

**7.3. JAVASCRIPT:**

* Javascript is a scripting languages, primarily used on the Websites.
* Javascript is used to enhance HTML pages and is commonly found embedded in HTML code.
* JavaScript is an interpreted language.
* JavaScript renders web pages in an interactive and dynamic fashion.

**Structure of Javascript**

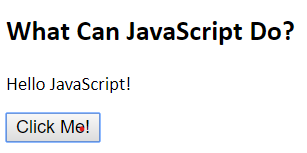
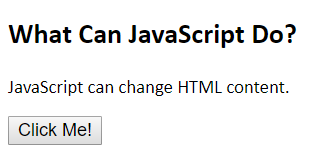
<!DOCTYPE html>  
<html>  
<head>

<h2>What Can JavaScript Do? </h2>

<p id="demo">JavaScript can change HTML content</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!“’ >Click Me!</button>

</body>  
</html>



**Example of Javascript**

<html>

<head></head>

<body>

<form onsubmit="return myfun()">

**Password** : <input type="password" id="passwords" value="">

<span id="messages" style="color: red;" placeholder="Password"> </span><br><br>

**Confirm Password** : <input type="password" id="passwordss" value="">

<span id="messagess" style="color: red;" placeholder="Confirm password"> </span><br><br>

<input type="submit" value="Submit">

</form>

</body>

<script>

function myfun() {

var a = document.getElementById("passwords").value;

var b = document.getElementById("passwordss").value;

if(a==""){

document.getElementById("messages").innerHTML="\*\*Please fill Password";

return false;

}

if(a.length < 5){

document.getElementById("messages").innerHTML="\*\*Password length must be greater than 5 characters";

return false;

}

if(a.length > 25){

document.getElementById("messages").innerHTML="\*\*Password length must be smaller than 25 characters";

return false;

}

if(b==""){

document.getElementById("messagess").innerHTML="\*\*Please fill Confirm Password";

return false;

}

if(b.length < 5){

document.getElementById("messagess").innerHTML="\*\*Confirm Password length must be greater than 5 characters";

return false;

}

if(b.length > 25){

document.getElementById("messagess").innerHTML="\*\*Confirm Password length must be smaller than 25 characters";

return false;

}

if(a!=b){

document.getElementById("messages").innerHTML="\*\*Password are not same!!!!";

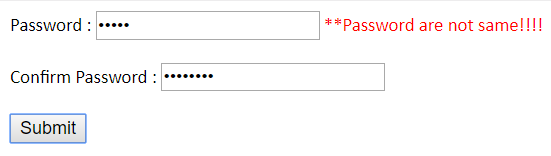
return false;

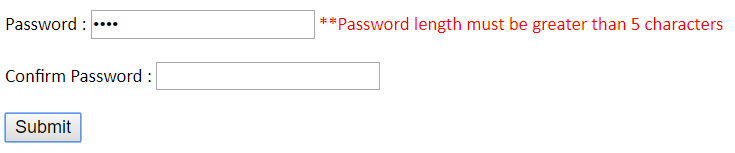
}

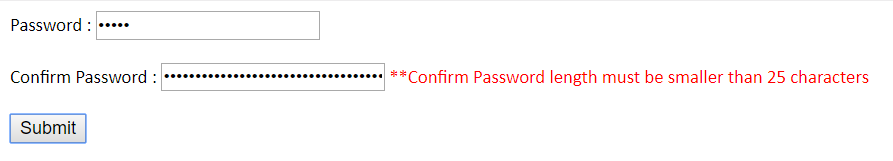
}

</script>

</html>







**7.4 Bootstrap:**

## What is Bootstrap? bootstrap.png

* Bootstrap is a free front-end framework for faster and easier web development
* Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
* Bootstrap also gives you the ability to easily create responsive designs
* **Bootstrap** is the most popular **CSS Framework** for developing responsive and mobile-first websites.
* **Bootstrap 4** is the newest version of Bootstrap



**Installation of Bootstrap**

<!DOCTYPE html>

<head>

<title>**Bootstrap Example**</title>

**<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">**

**<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>**

**<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>**

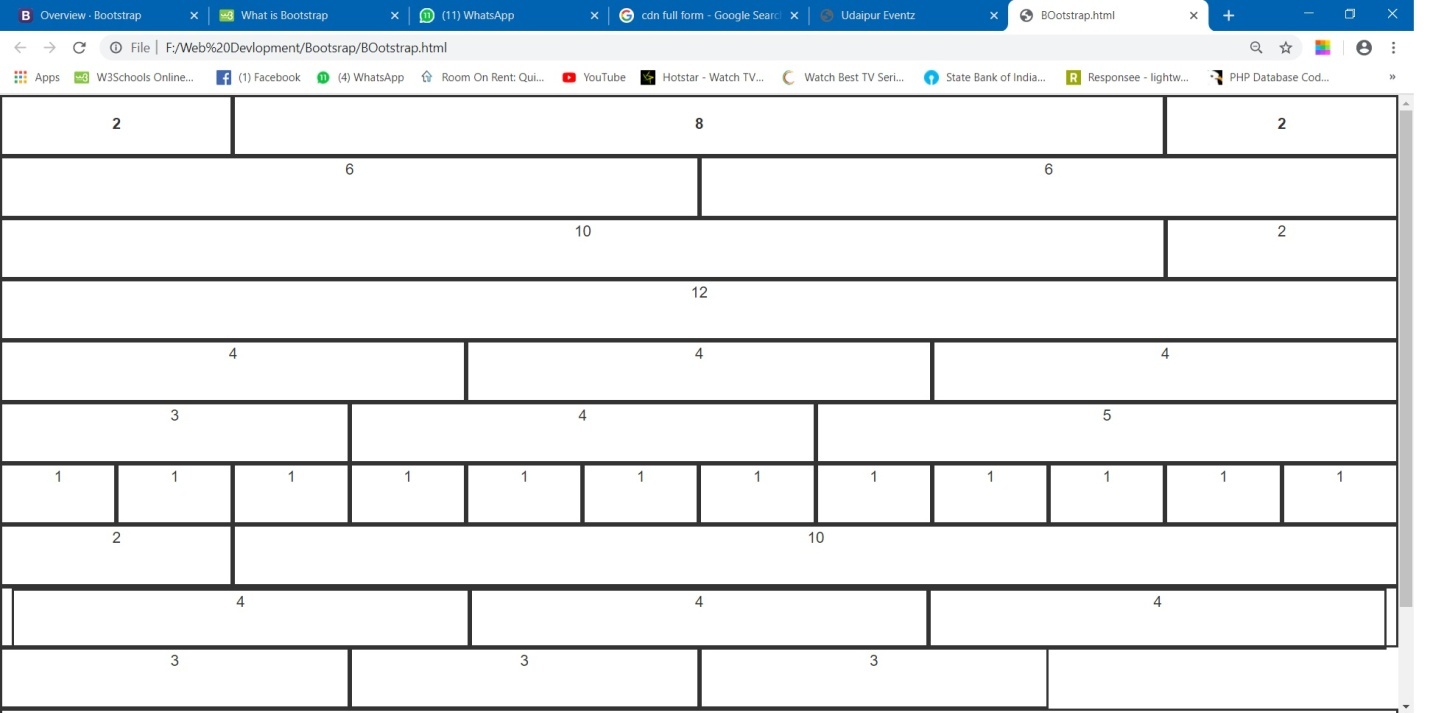
**<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>**

</head>

</html>

**Grid System**

* Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.
* If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:



* The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.
* Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

**Grid Classes**

The Bootstrap 4 grid system has five classes:

* **.col-** (extra small devices - screen width less than 576px)
* **.col-sm-**

(small devices - screen width equal to or greater than 576px)

* **.col-md-**

 (medium devices - screen width equal to or greater than 768px)

* **.col-lg-**

 (large devices - screen width equal to or greater than 992px)

* **.col-xl-**

(xlarge devices - screen width equal to or greater than 1200px)

The classes above can be combined to create more dynamic and flexible layouts.

## Basic Structure of a Bootstrap 4 Grid

**<div class="continer" style= "height:auto ;border:none">**

<div class="col-lg-2 col-md-2 col-sm-2 col-xs-2" ><label>2</label></div>

<div class="col-lg-8 col-md-8 col-sm-8 col-xs-8" ><label>8</label></div>

<div class="col-lg-2 col-md-2 col-sm-2 col-xs-2" ><label>2</label></div>

<div class="col-lg-6 col-md-6 col-sm-6 col-xs-6" >6</div>

<div class="col-lg-6 col-md-6 col-sm-6 col-xs-6" >6</div>

<div class="col-lg-10 col-md-10 col-sm-10 col-xs-10" >10</div>

<div class="col-lg-2 col-md-2 col-sm-2 col-xs-2" >2</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12" >12</div>

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

<div class="col-lg-3 col-md-3 col-sm-3 col-xs-3" >3</div>

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

<div class="col-lg-5 col-md-5 col-sm-5 col-xs-5" >5</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<div class="col-lg-1 col-md-1 col-sm-1 col-xs-1" >1</div>

<br><br><br><br><br>

<div class="col-lg-2 col-md-2 col-sm-2 col-xs-2" >2</div>

<div class="col-lg-10 col-md-10 col-sm-10 col-xs-10" >10</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12" >

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4" >4</div>

</div>

<div class="col-lg-3 col-md-3 col-sm-3 col-xs-3" >3</div>

<div class="col-lg-3 col-md-3 col-sm-3 col-xs-3" >3</div>

<div class="col-lg-3 col-md-3 col-sm-3 col-xs-3" >3</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12" >12</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12" >12</div>

**</div>**

**Media Query v/s Bootstrap**

## Media Queries Simple Examples

* One way to use media queries is to have an alternate CSS section right inside your style sheet.
* The following example changes the background-color to lightgreen if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the background-color will be pink):

**<style type="text/css">**

**@media screen and (min-width: 550px){**

**.myimg {**

**width:720px !important;**

**}**

**}**

**@media screen and (min-width: 1530px){**

**.myimg {**

728x90px

**width:620px !important;**

**}**

**}**

**@media screen and (min-width: 1494px ){**

**.myimg {**

**width:620px !important;**

300 x 250px

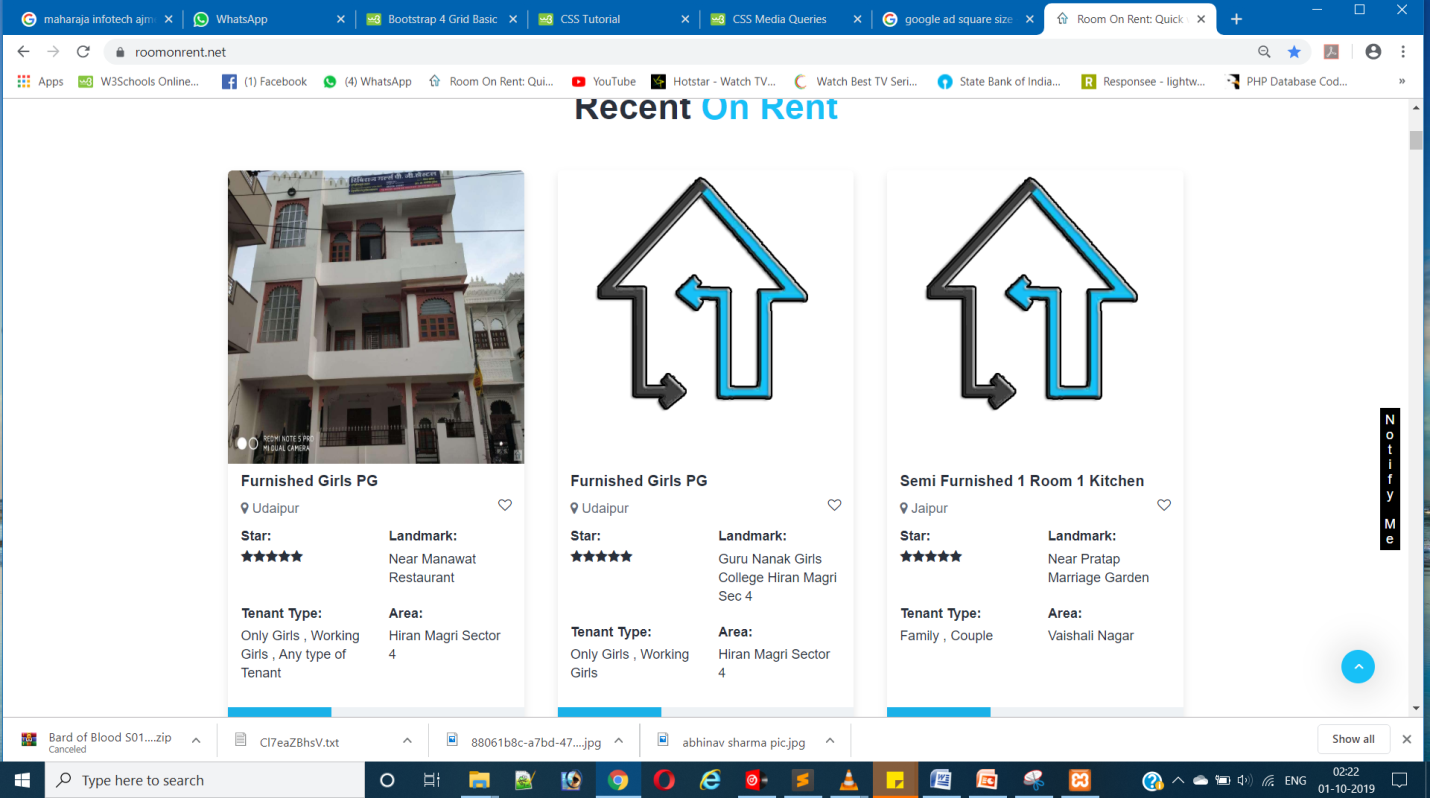
**}**

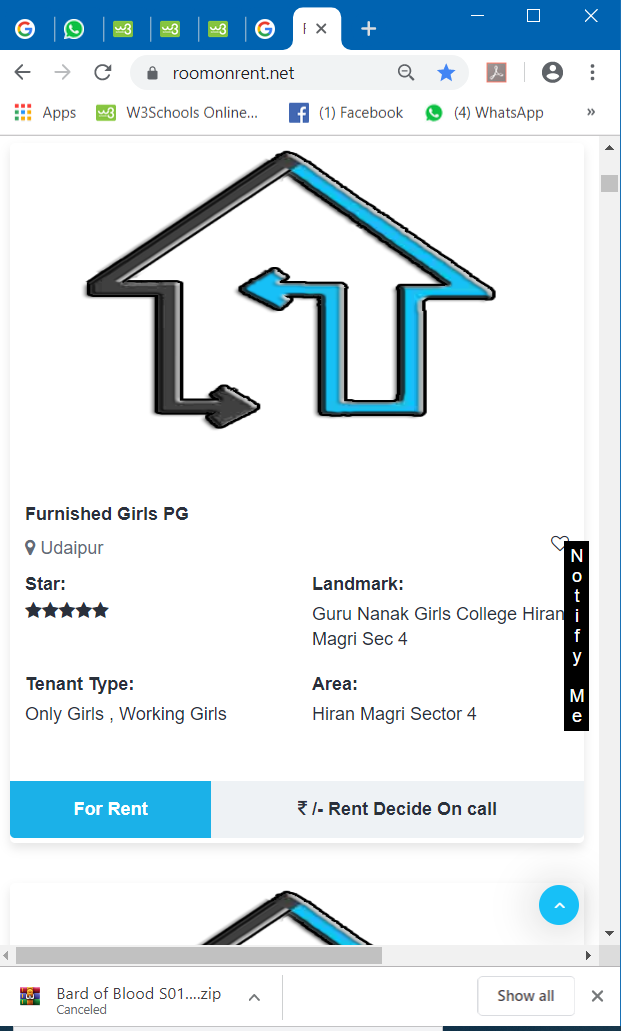
**}**

**</style>**

**<section>**

**Bootstrap Example**





**7.5 SQL:** SQL is a standard language for storing, manipulating and retrieving data in databases.

SQL stands for Structured Query Language.

**What Can SQL do?**

SQL can **execute** queries against a database

SQL can **create** new databases

SQL can **retrieve** data from a database

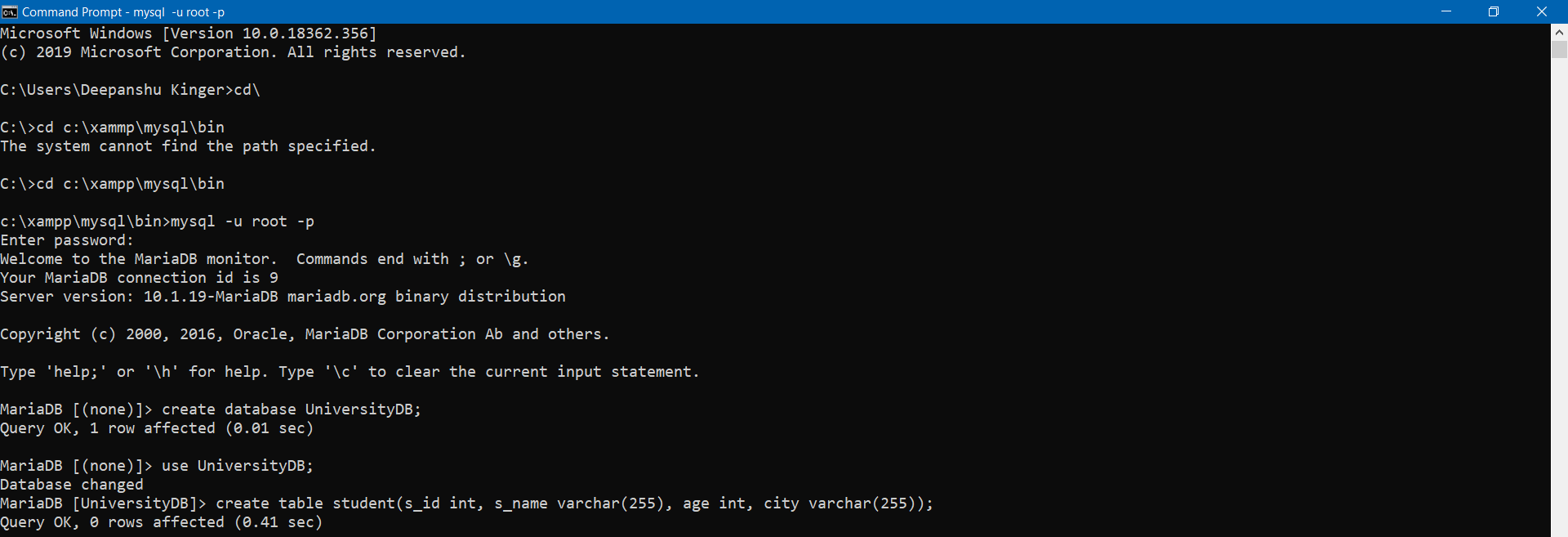
SQL can **insert** records in a database

SQL can **update** records in a database

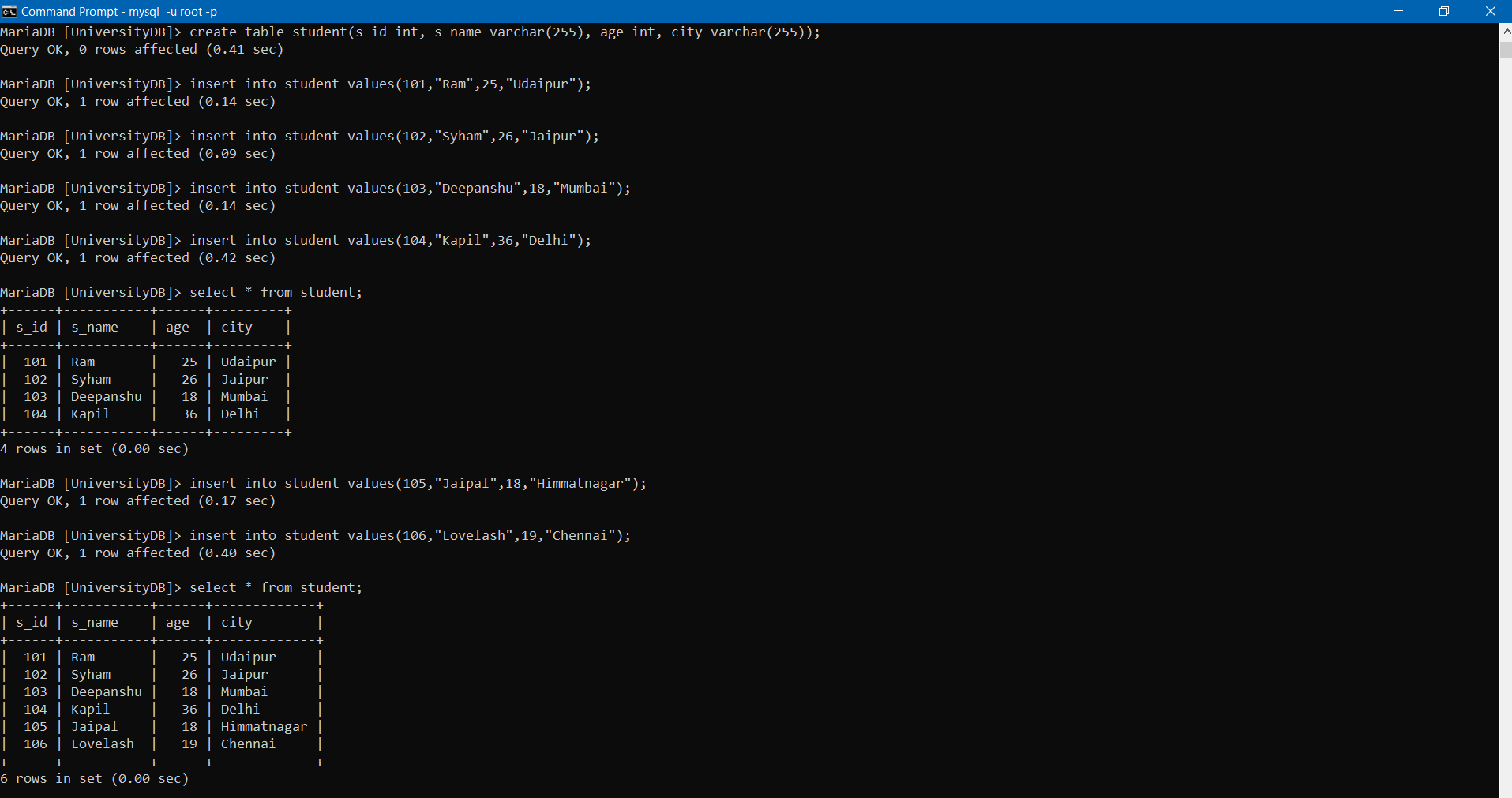
SQL can **delete** records from a database

SQL can **create** new tables in a database

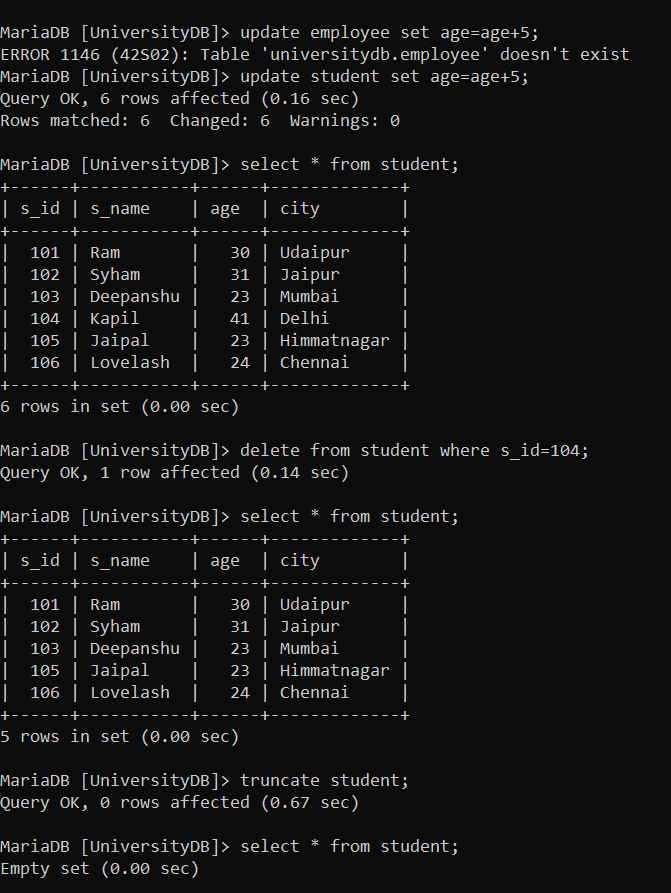
**Basic Queries of SQL**

**

**Insert in SQL**

**

**Update , Delete in SQ**

**L**

**7.6 PHP: Hypertext Preprocessor**

* Open source Scripting language.
* Scripts are executed on the SERVER.
* PHP files have extension ".php"

**What Can PHP Do?**

* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can be used to control

## php-introduction-01-splessons.jpg

## Why PHP?

* PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP supports a wide range of databases
* PHP is free. Download it from the official PHP resource: [www.php.net](http://www.php.net/)
* PHP is easy to learn and runs efficiently on the server side

## What's new in PHP 7

* PHP 7 is much faster than the previous popular stable release (PHP 5.6)
* PHP 7 has improved Error Handling

**Syntax of PHP**

Placed anywhere in the document.

A PHP script starts with **<?php** and ends with **?>**:

**<?php**  
// PHP code goes here

echo "Hello World!";  
**?>**