Module 1: Software Engineering, HTML and CSS

1. (a) Create a html web page that contains a button with which it is possible to change the text that is shown on the screen. The file ButtonDemo.html has a JavaScript function named change\_text() which is called after the button is pressed. When the button is pressed repeatedly the text changes Hello! ... Well done! ... Hello! ... Well done! ... Hello! ..

1.=====Code

<html>

<script>

function change\_text()

{

var text=document.createTextNode("Hello!...well done!...");

var p=document.createElement(p);

var temp=p.appendChild(text);

document.body.appendChild(temp);

}

</script>

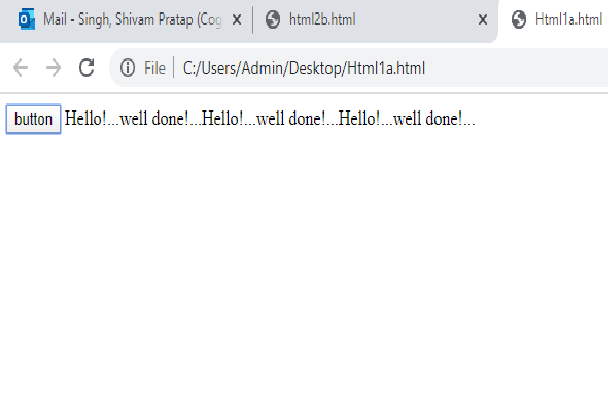
<body>

<input type="submit" value="button" onclick="change\_text()">

</body>

</html>

1.Output ScreenShot



(b) Modify the program so that the initial text shown on the screen is "Monday", and it will change in the following way when the button is

pressed repeatedly: Monday ... Tuesday ... Wednesday ... Thursday ... Friday ... Saturday ... Sunday ... Monday ... Tuesday ... etc. You should also change the button text so that there is written "Change day" on the button.

(b)….. CODE////

<!DOCTYPE html>

<html>

<script>

var index=0;

function change\_day()

{

if (index>6)

{

index=0;

}

var days=['Monday ... ','Tuesday ... ','Wednesday ... ','Thursday ... ','Friday ... ','Saturday ... ','Sunday ... '];

var text=document.createTextNode(days[index]);

var p=document.createElement(p);

var temp=p.appendChild(text);

document.body.appendChild(temp);

index=index +1;

}

</script>

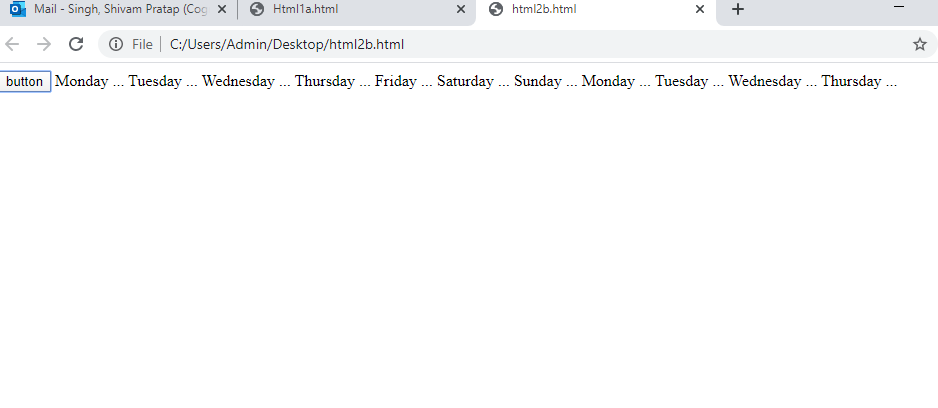
<body>

<input type="submit" value="button" onclick="change\_day()">

</body>

</html>

(b) output//////// screenshot



3. Using CSS properties create a calendar with following Calender Template:

3.CODE/////

<!DOCTYPE html>

<html>

<head>

<style>

\* {box-sizing: border-box;}

ul {list-style-type: none;}

body {font-family: Verdana, sans-serif;

background-image: url("cssimg.jpg");

}

.month {

padding: 10px 25px;

width: 100%;

background: red;

text-align: center;

}

.month ul {

margin: 1

padding: 0;

}

.month ul li {

color: white;

font-size: 30px;

text-transform: uppercase;

letter-spacing: 3px;

}

.month .prev {

float: left;

padding-top: 10px;

}

.month .next {

float: right;

padding-top: 5px;

}

.weekdays {

margin: 0;

padding: 10px 0;

background-color: white;

}

.weekdays li {

display: inline-block;

width: 13.6%;

color: #666;

text-align: center;

}

.days {

padding: 10px 0;

background: #eee;

margin: 0;

}

.days li {

list-style-type: none;

display: inline-block;

width: 13.6%;

text-align: center;

margin-bottom: 5px;

font-size:12px;

color: blue;

}

.days li .active {

padding: 5px;

background: #1abc9c;

color: white !important

}

</style>

</head>

<body >

<h1 align="center">CSS Calendar</h1>

<div class="month">

<ul>

<li>

January<br>

<span style="font-size:18px">2020</span>

</li>

</ul>

</div>

<ul class="weekdays">

<li>Mo</li>

<li>Tu</li>

<li>We</li>

<li>Th</li>

<li>Fr</li>

<li>Sa</li>

<li>Su</li>

</ul>

<ul class="days">

<li>1</li>

<li>2</li>

<li>3</li>

<li>4</li>

<li>5</li>

<li>6</li>

<li>7</li>

<li>8</li>

<li>9</li>

<li>10</li>

<li>11</li>

<li>12</li>

<li>13</li>

<li>14</li>

<li>15</li>

<li>16</li>

<li>17</li>

<li>18</li>

<li>19</li>

<li>20</li>

<li>21</li>

<li>22</li>

<li>23</li>

<li>24</li>

<li>25</li>

<li>26</li>

<li>27</li>

<li>28</li>

<li><span class="active">29</span></li>

<li>30</li>

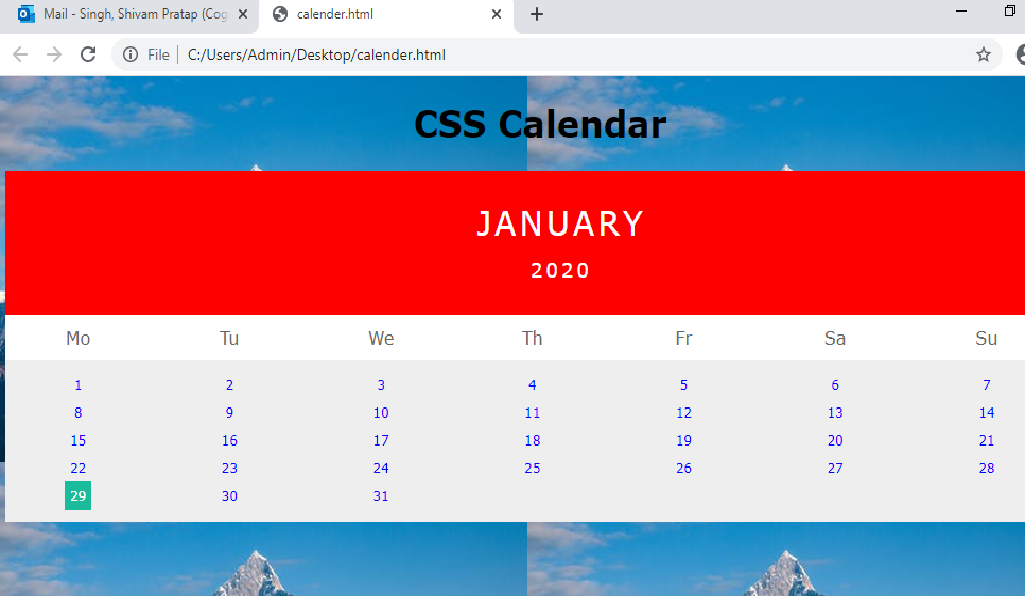
<li>31</li>

</ul>

</body>

</html>

==============OUTPUT- SCREENSHOT======



Module 2: Java Script and SQL

4. Write the queries Based on following Loan Schema:

Queries:

a. Write a query to display customer number, customer’s firstname , account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.

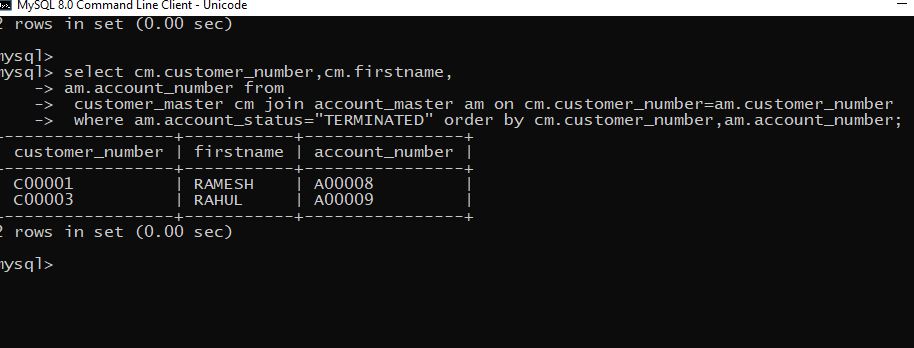
Sqlquery==========

select cm.customer\_number,cm.firstname,

am.account\_number from

customer\_master cm join account\_master am on cm.customer\_number=am.customer\_number

where am.account\_status="TERMINATED" order by cm.customer\_number,am.account\_number;



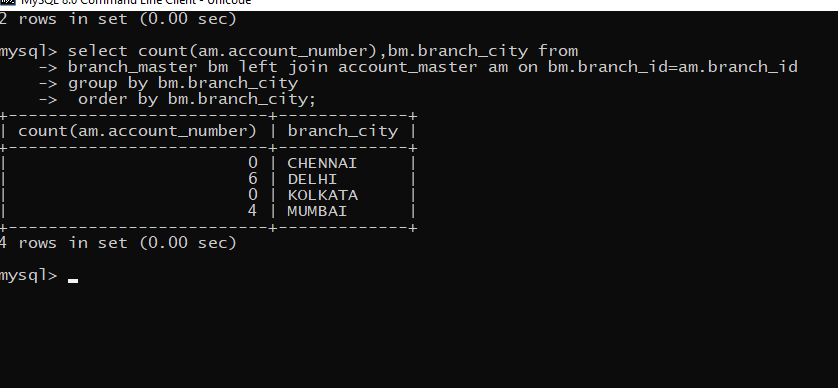
b. Write a query to display the number of accounts opened in each city. The query should display the Branch city and the number of No\_of\_Accounts for the branch city where we don’t have any accounts opened display 0. Display the records in the sorted order based on branch city

select count(am.account\_number),bm.branch\_city from

branch\_master bm left join account\_master am on bm.branch\_id=am.branch\_id

group by bm.branch\_city

order by bm.branch\_city;



c. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.

select cm.customer\_number,cm.firstname,cm.lastname

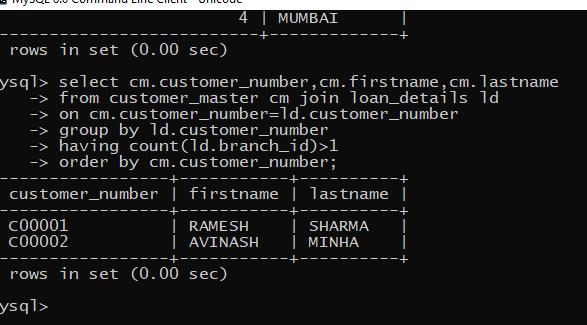
from customer\_master cm join loan\_details ld

on cm.customer\_number=ld.customer\_number

group by ld.customer\_number

having count(ld.branch\_id)>1

order by cm.customer\_number;



d. Write a query to display the number of clients who have asked for loan but they don’t have any account in the bank though they are registered customers. Give the count an alias name of Count

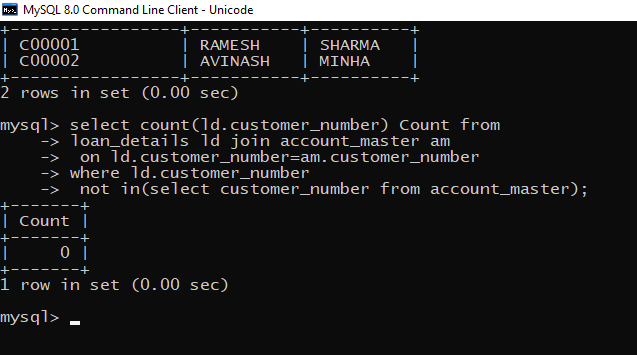
select count(ld.customer\_number) Count from

loan\_details ld join account\_master am

on ld.customer\_number=am.customer\_number

where ld.customer\_number

not in(select customer\_number from account\_master);



e. Write a equerry to display customers firstname, city and account number whose occupation are not business , Services or Student. Display the records sorted in ascending order based on customer firstname and by account number.

select cm.firstname,cm.customer\_city,am.account\_number

from account\_master am join customer\_master cm on am.customer\_number=cm.customer\_number

where cm.customer\_number NOT IN

(select customer\_number from customer\_master where occupation ='SERVICE' or 'STUDENT' or 'BUSINESS' )

order by cm.firstname,am.account\_number;

