**Module 2: Java Script and SQL**

1. Write a guessing game where the user has to guess a secret number. After every guess the program tells the user whether their number was too large or too small. At the end the number of tries needed should be printed. It counts only as one try if they input the same number multiple times consecutively.

<html>

<head>

<script>

var count=0;

function guessNumber(){

count++;

var val=document.getElementById("num").value;

var number=80078;

if(val<number)

{

document.writeln("number is too small");

}

else if(val>number){

document.writeln("number is too large");

}

else

{

document.writeln("guessing number is correct");

}

document.writeln("number of tries "+count)

}

</script>

</head>

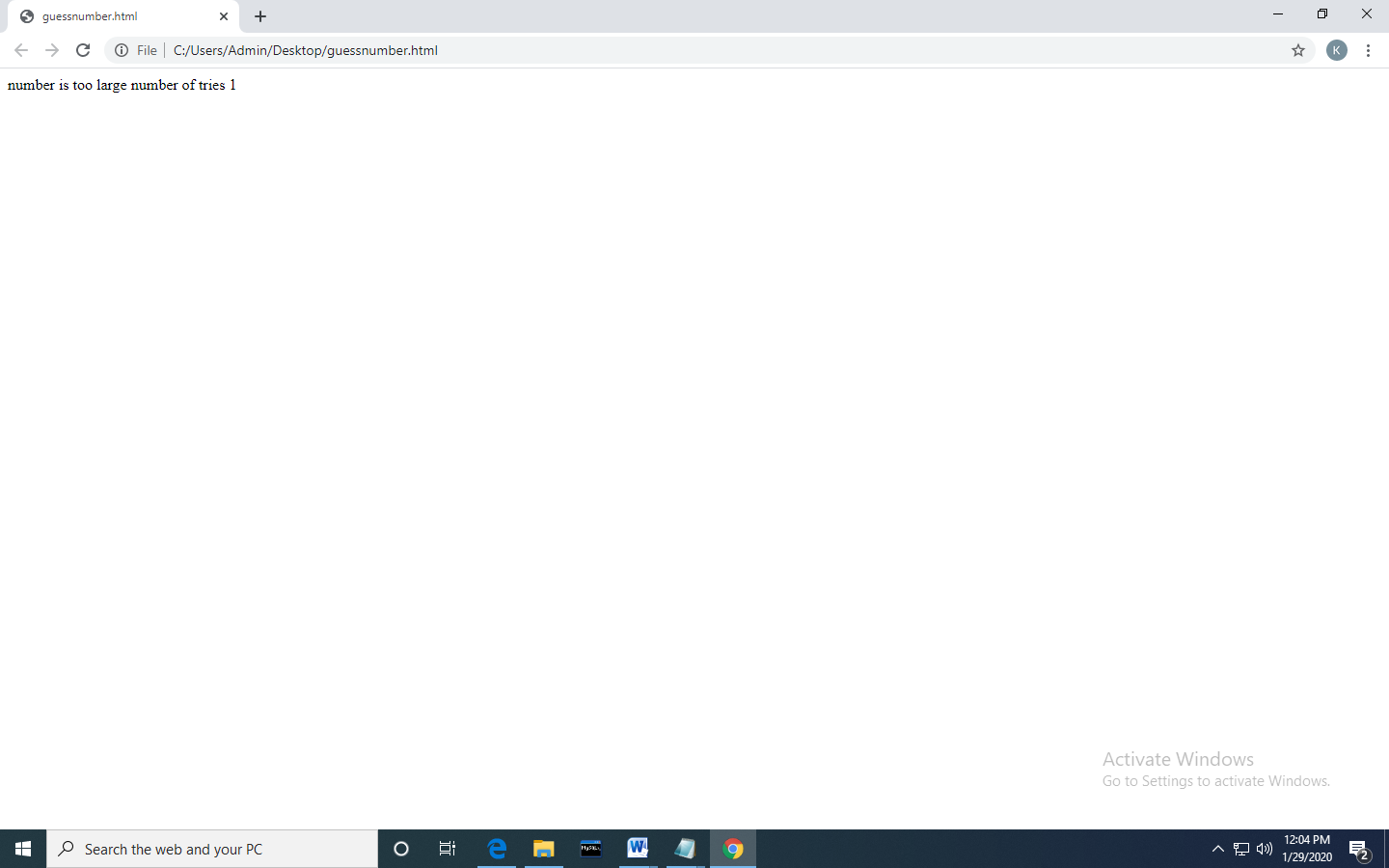
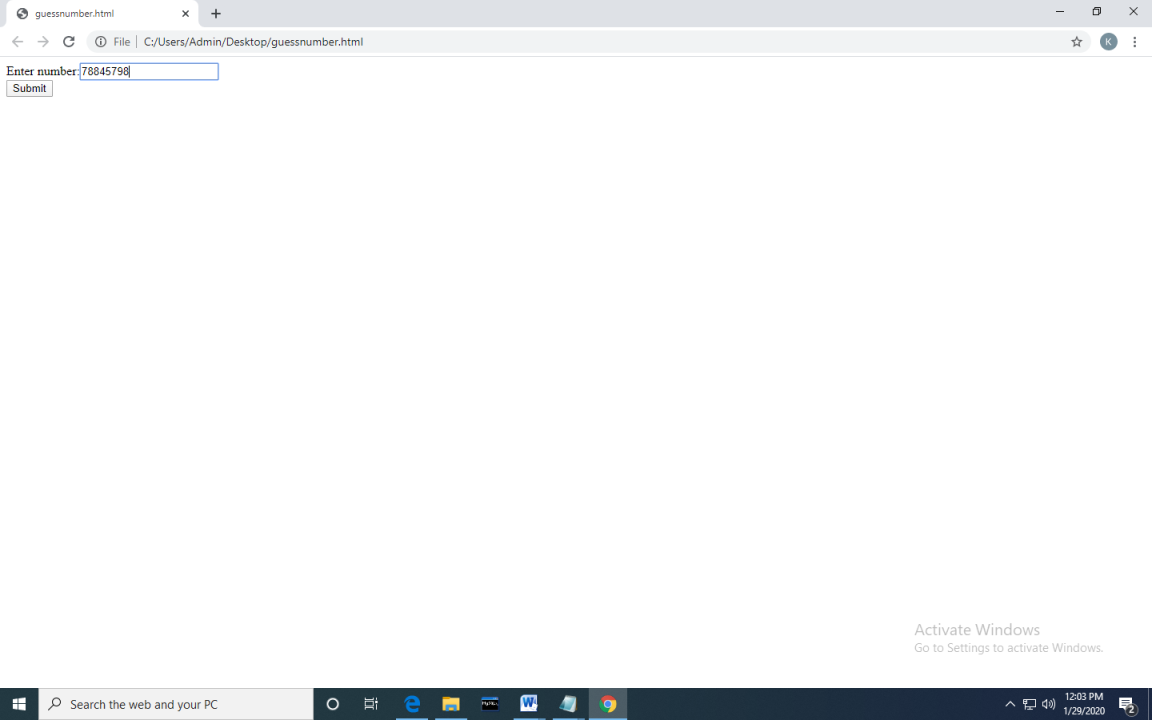
<body>

Enter number:<input type='text' id='num'><br>

<input type='submit' onclick='guessNumber()'>

</body>

</html>



2.Write a Java Script to find the n prime number.

<!DOCTYPE html>

<html>

<head>

<script>

function nextPrime(value) {

if (value > 2) {

var i, q;

do {

i = 3;

value += 2;

q = Math.floor(Math.sqrt(value));

while (i <= q && value % i) {

i += 2;

}

} while (i <= q);

return value;

}

return value === 2 ? 3 : 2;

}

function prime() {

var userValue = document.getElementById("inputValue").value;

var value = 0, result = [];

for (var i = 0; i < userValue; i++) {

value = nextPrime(value);

result.push(value);

}

document.getElementById("returnValue").innerHTML = result[userValue-1];

}

</script>

</head>

<body>

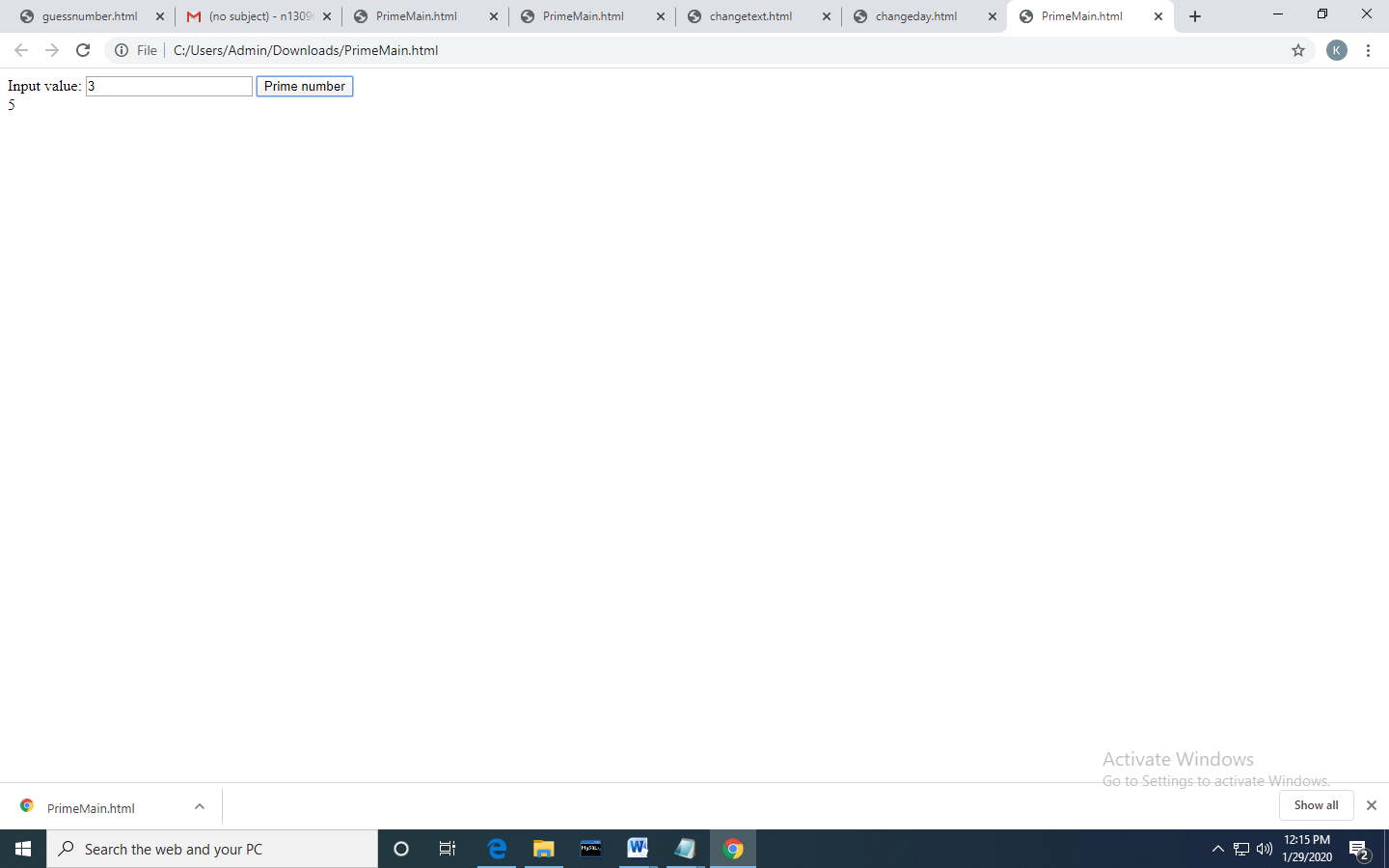
Input value: <input type="text" name="inputValue" id="inputValue"/>

<button onclick="prime()">Prime number</button>

<div id="returnValue">Test: </div>

</body>

</html>

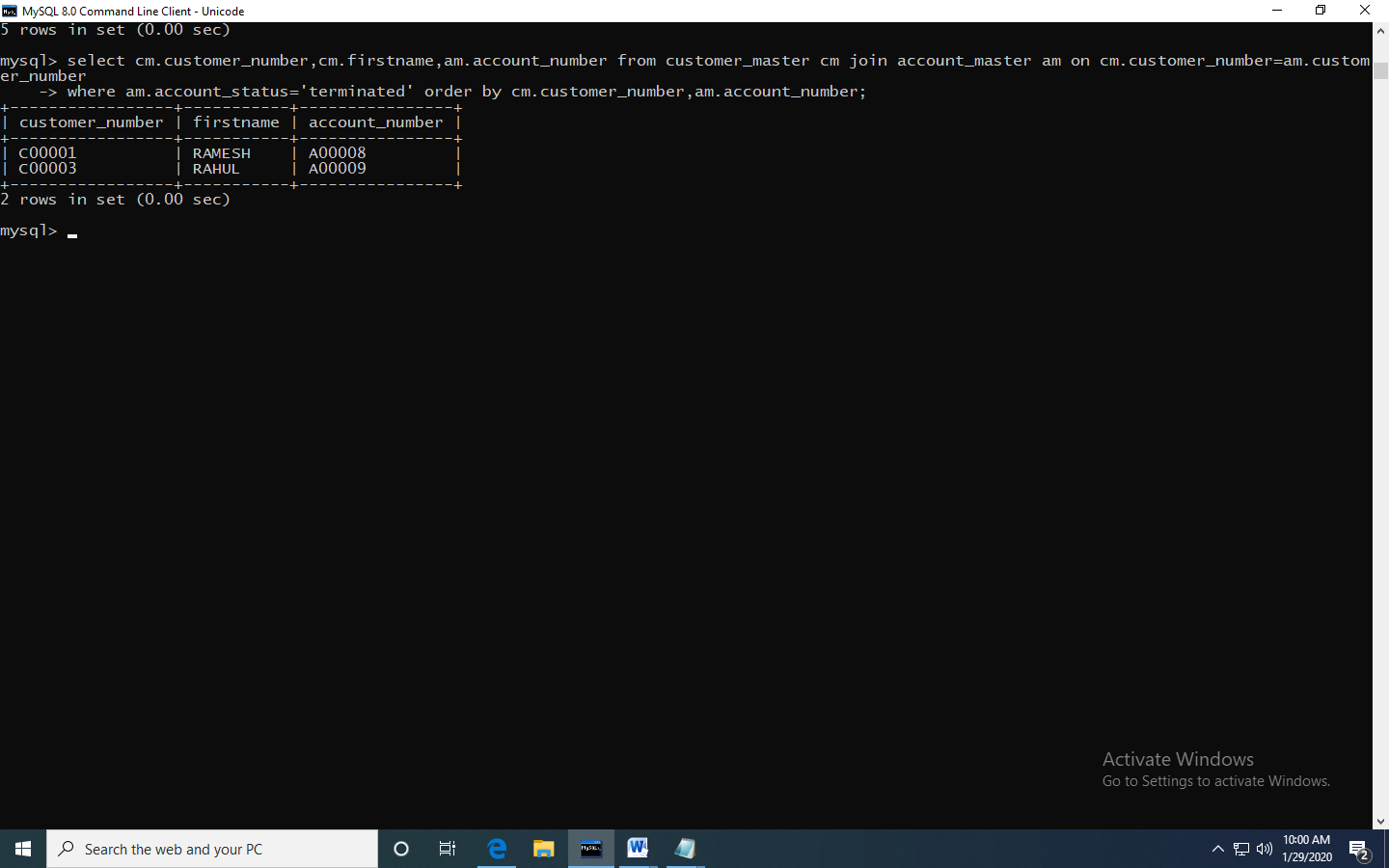


4.Write the queries Based on following Loan Schema:

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.

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;

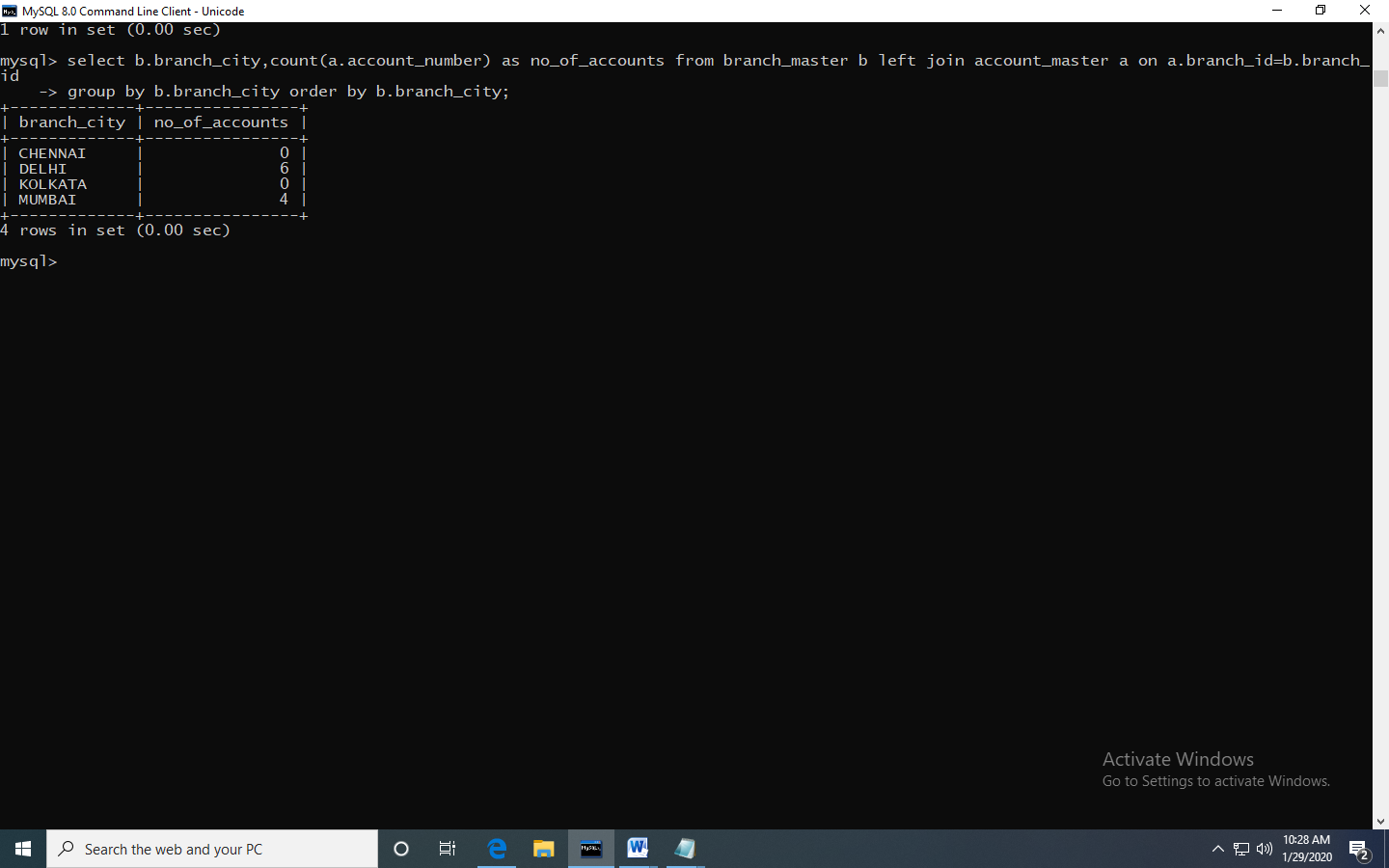


b.

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 b.branch\_city,count(a.account\_number) as no\_of\_accounts from branch\_master b left join account\_master a on a.branch\_id=b.branch\_id

group by b.branch\_city order by b.branch\_city;

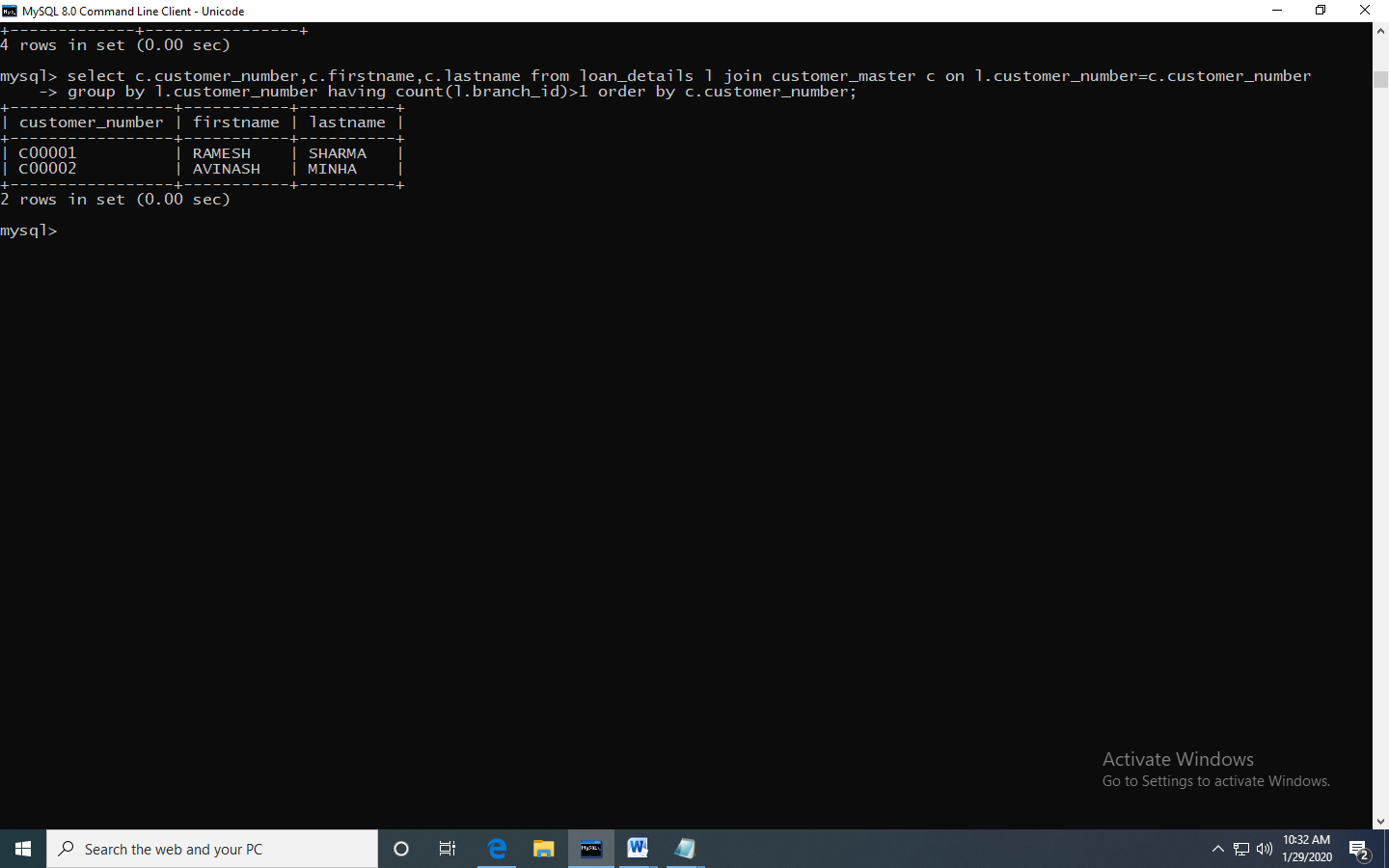


C.

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 c.customer\_number,c.firstname,c.lastname from loan\_details l join customer\_master c on l.customer\_number=c.customer\_number

group by l.customer\_number having count(l.branch\_id)>1 order by c.customer\_number;

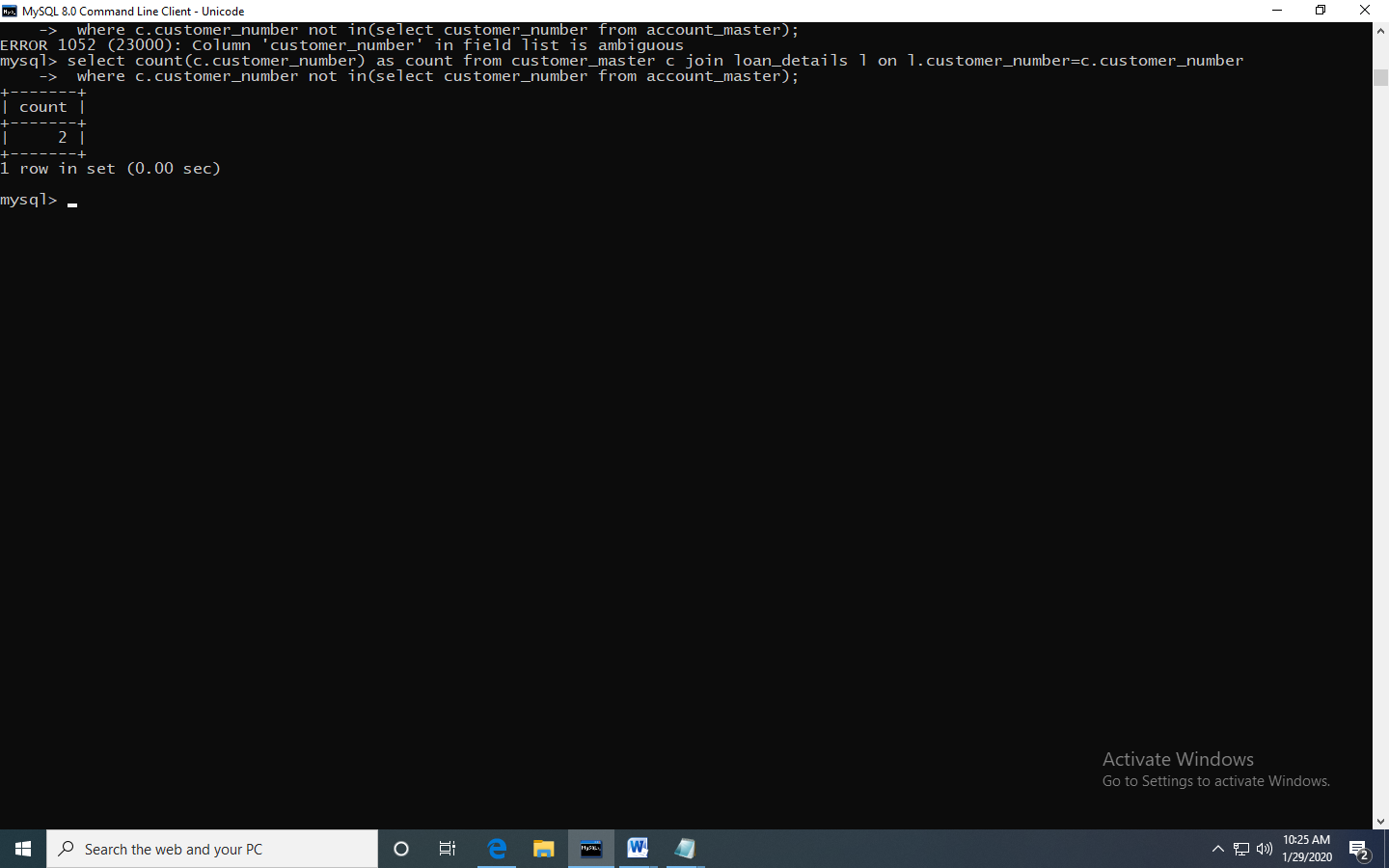


d.

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(c.customer\_number) as count from customer\_master c join loan\_details l on l.customer\_number=c.customer\_number

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

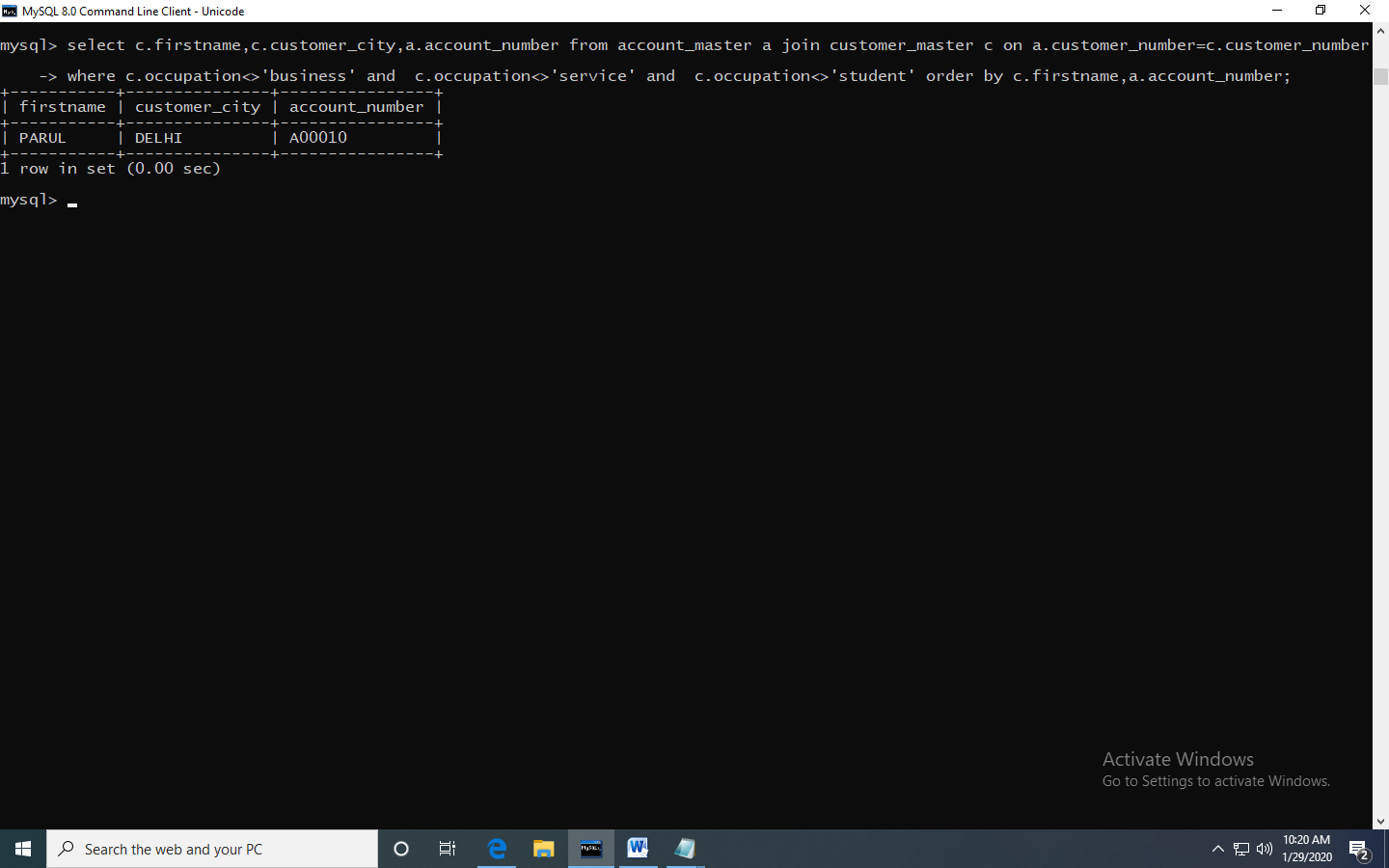


e.

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 c.firstname,c.customer\_city,a.account\_number from account\_master a join customer\_master c on a.customer\_number=c.customer\_number

where c.occupation<>'business' and c.occupation<>'service' and c.occupation<>'student' order by c.firstname,a.account\_number;



**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! ..

<html>

<head>

<script>

var pos=0;

function change\_text(){

if(pos==0){

document.getElementById("d1").innerHTML="Weldone!....";

pos=1;

}

else{

document.getElementById("d1").innerHTML="Hello....";

pos=0;

}

}

</script>

</head>

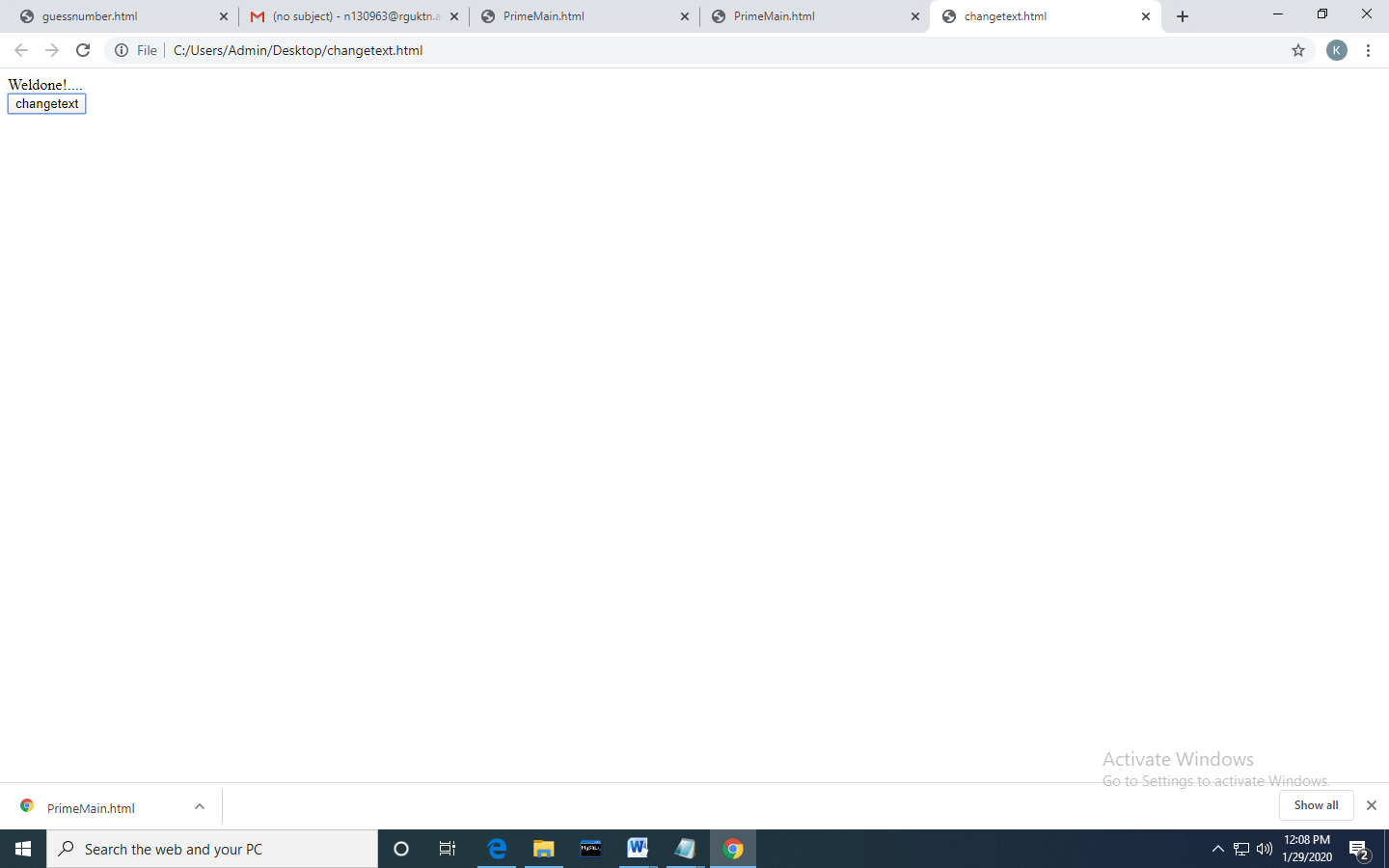
<body>

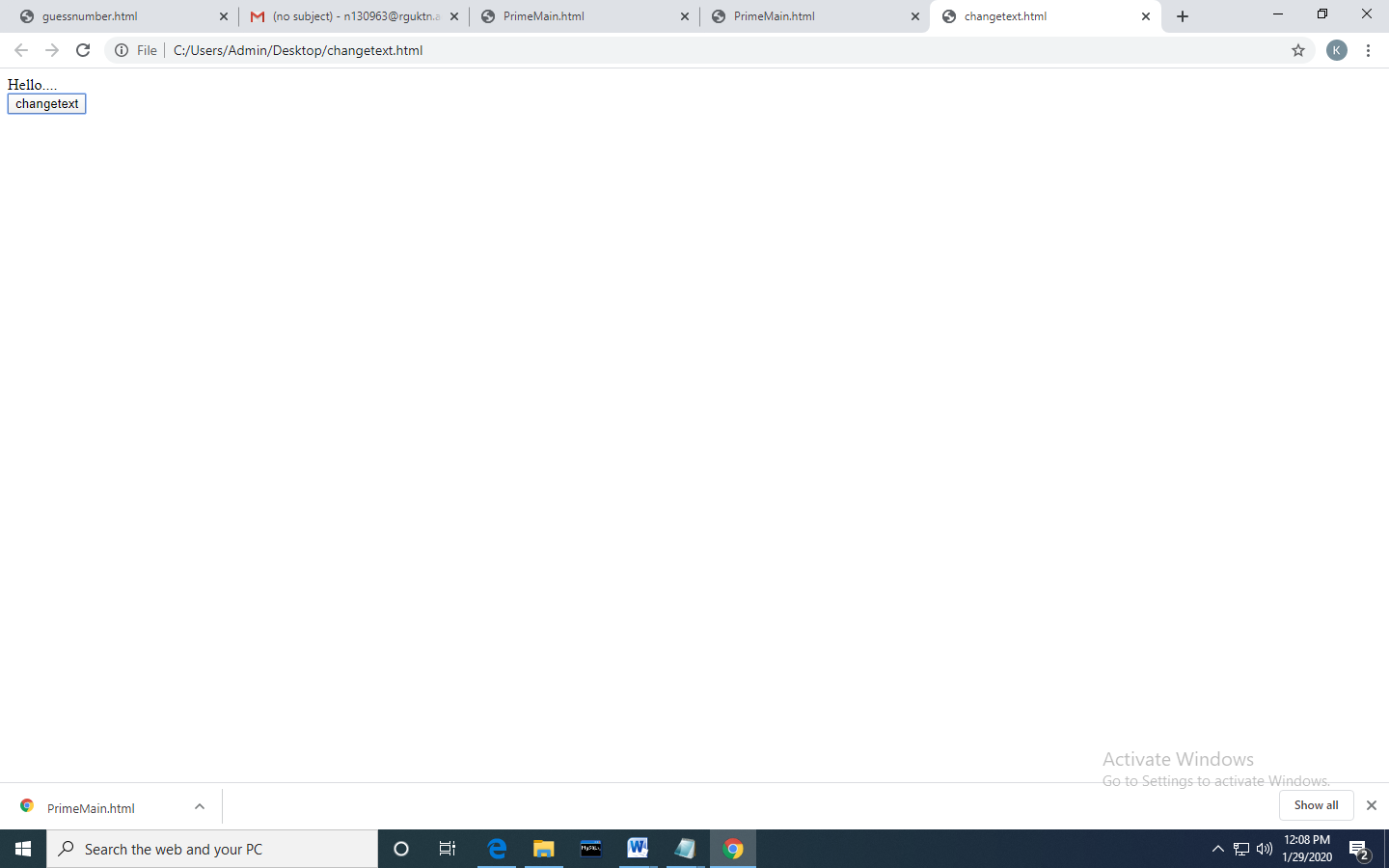
<div id="d1">Hello</div>

<input type="button" id="b1" value="changetext" onclick="change\_text()">

</body>

</html>





(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.

<html>

<head>

<script>

function change\_day(){

var val=document.getElementById("d1");

if(val.innerHTML=== "Sunday"){

val.innerHTML="Monday";

}

else if(val.innerHTML=== "Monday"){

val.innerHTML="Tuesday";

}

else if(val.innerHTML=== "Tuesday"){

val.innerHTML="Wednesday";

}

else if(val.innerHTML=== "Wednesday"){

val.innerHTML="Thrusday";

}

else if(val.innerHTML=== "Thrusday"){

val.innerHTML="Friday";

}

else if(val.innerHTML=== "Friday"){

val.innerHTML="Saturday";

}

else {

val.innerHTML="Sunday";

}

}

</script>

</head>

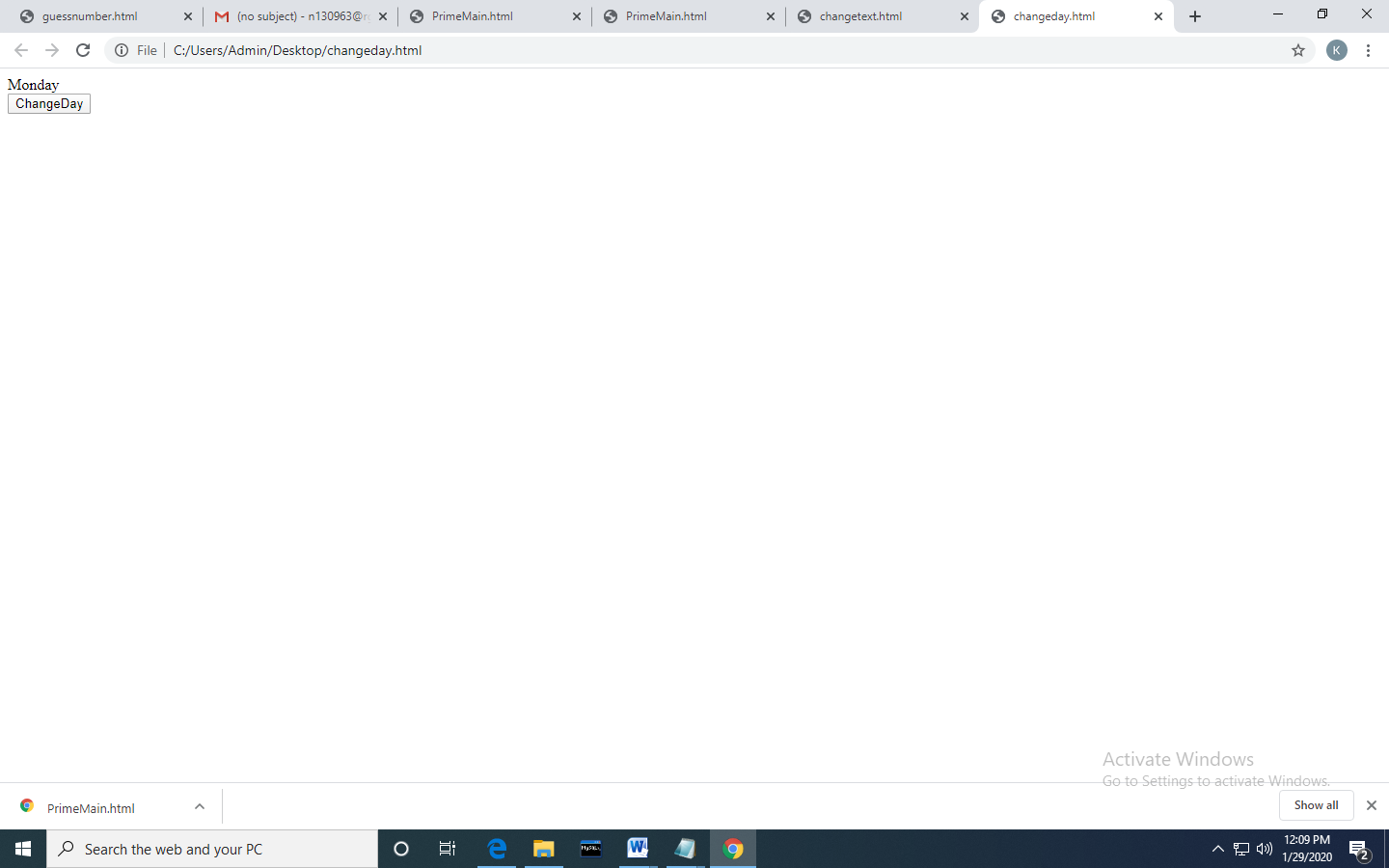
<body>

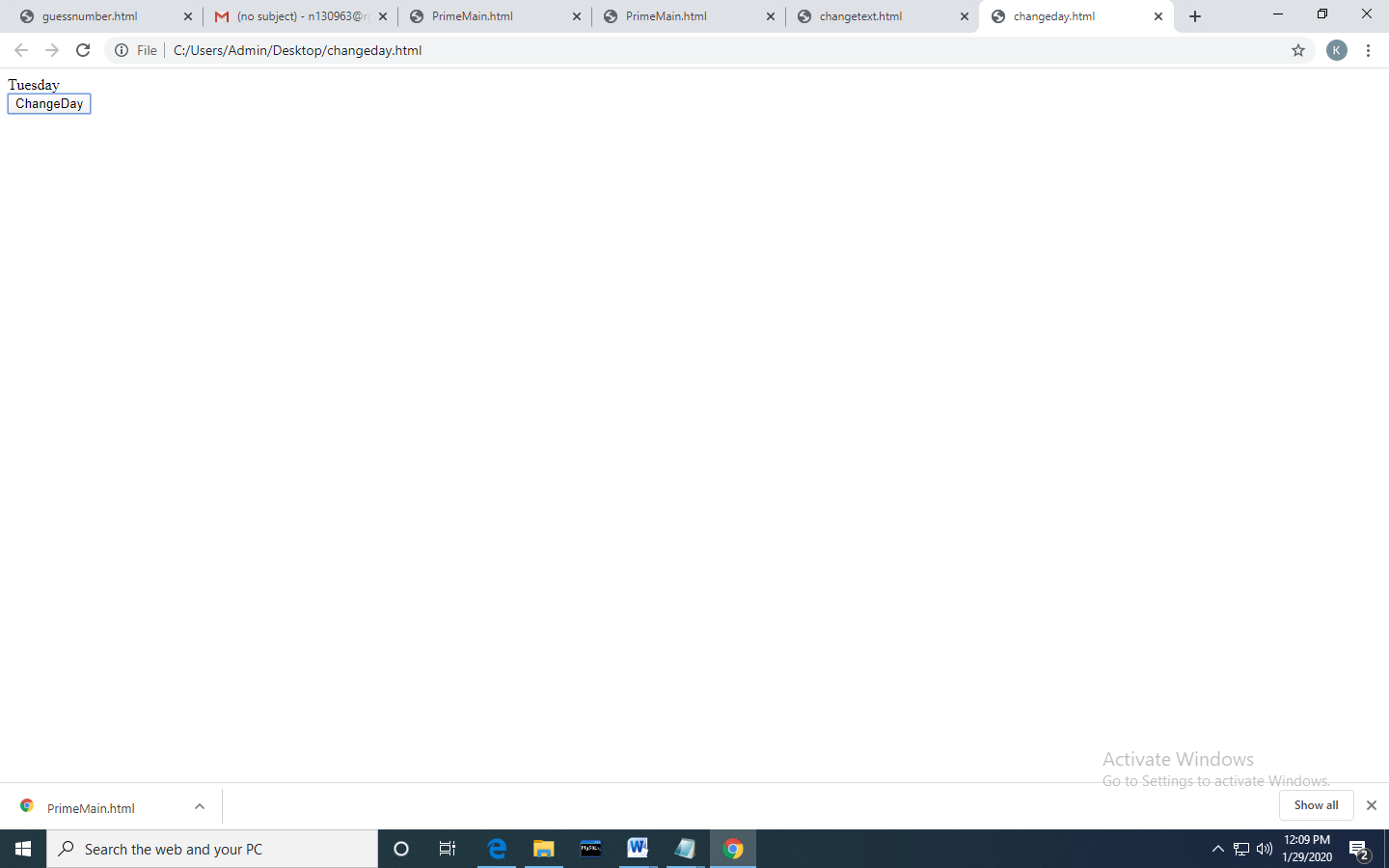
<div id="d1" >Monday</div>

<input type="button" id="b1" value="ChangeDay" onclick="change\_day()">

</body>

</html>





2

a. Create a newHTML-document with an unordered list element, a text-box,and a button that says “Add.”

b. Add JavaScript(and/or jQuery)that appendsa new <li> element to the unordered list when the button is clicked. The text of the new li-element should correspond to the text entered by the user in the text-box. Make sure thatthe content of the text-box is cleared when the button is clicked to be ready for new input from the user.

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<div id="d1" class="header">

<input type="text" id="myInput" placeholder="Add">

<button onclick="newElement()" class="addBtn">Add</button>

</div>

<ul id="u1">

</ul>

<script>

var myNodelist = document.getElementsByTagName("LI");

var i;

for (i = 0; i < myNodelist.length; i++) {

var span = document.createElement("SPAN");

var txt = document.createTextNode("\u00D7");

span.className = "close";

span.appendChild(txt);

myNodelist[i].appendChild(span);

}

var list = document.querySelector('ul');

list.addEventListener('click', function(ev) {

if (ev.target.tagName === 'LI') {

ev.target.classList.toggle('checked');

}

}, false);

function newElement() {

var li = document.createElement("li");

var inputValue = document.getElementById("myInput").value;

var t = document.createTextNode(inputValue);

li.appendChild(t);

if (inputValue === '') {

} else {

document.getElementById("u1").appendChild(li);

}

document.getElementById("myInput").value = "";

var span = document.createElement("SPAN");

var txt = document.createTextNode("\u00D7");

span.className = "close";

span.appendChild(txt);

li.appendChild(span);

for (i = 0; i < close.length; i++) {

close[i].onclick = function() {

var div = this.parentElement;

div.style.display = "none";

}

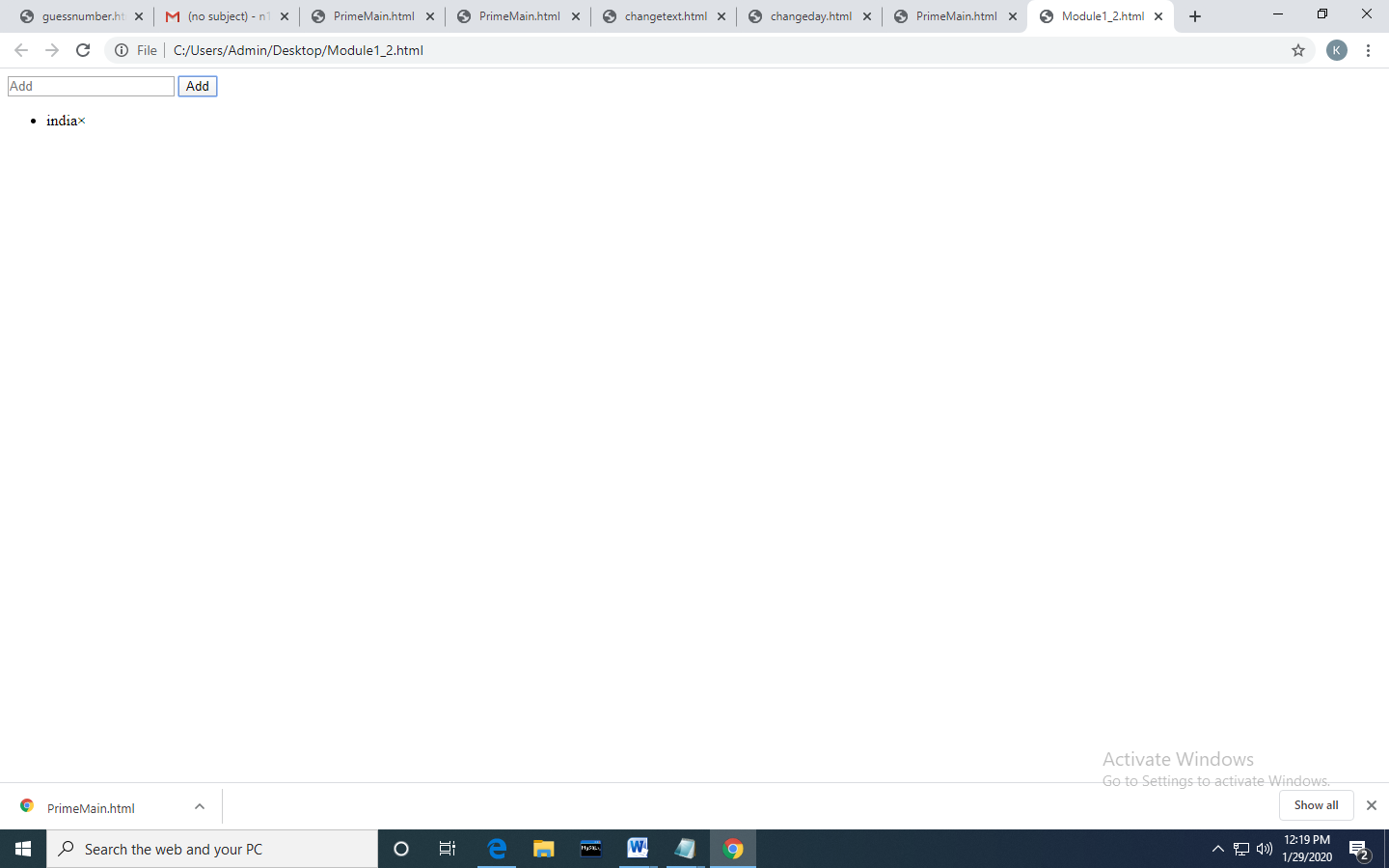
}

}

</script>

</body>

</html>



1. Using CSS properties create a calendar with following Calender Template

<html>

<head>

<style>

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

ul {list-style-type: none;}

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

.month {

padding: 5px 25px;

width: 100%;

background:skyblue;

text-align: center;

}

.month ul {

margin: 0;

padding: 0;

}

.month ul li {

color: white;

font-size: 20px;

text-transform: uppercase;

letter-spacing: 3px;

}

.month .prev {

float: left;

padding-top: 10px;

}

.month .next {

float: right;

padding-top: 10px;

}

.weekdays {

margin: 0;

padding: 10px 0;

background-color: #ddd;

}

.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: #777;

}

.days li .active {

padding: 5px;

background:#f0f0f0;

color: white !important

}

/\* Add media queries for smaller screens \*/

@media screen and (max-width:720px) {

.weekdays li, .days li {width: 13.1%;}

}

@media screen and (max-width: 420px) {

.weekdays li, .days li {width: 12.5%;}

.days li .active {padding: 2px;}

}

@media screen and (max-width: 290px) {

.weekdays li, .days li {width: 12.2%;}

}

</style>

</head>

<body>

<h1>Calendar</h1>

<div class="month">

<ul>

<li class="prev">&#10094;</li>

<li class="next">&#10095;</li>

<li>

MAY<br>

<span style="font-size:18px">2017</span>

</li>

</ul>

</div>

<ul class="weekdays">

<li>Mon</li>

<li>Tue</li>

<li>Wed</li>

<li>Thu</li>

<li>Fri</li>

<li>Sat</li>

<li>Sun</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><span class="active">15</span></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>29</li>

<li>30</li>

<li>31</li>

</ul>

</body>

</html>

