

■ Introduction

WAP is an international standard establishing how mobile devices can access information on the Internet. It is a widely used set of protocols used on wireless devices such as mobile phones

WAP is the worldwide standard for providing Internet communications and advanced telephony services on digital mobile phones, pagers, personal digital assistants, and other wireless terminals – *WAP Forum*.

WAP stands for **Wireless Application Protocol**. The dictionary definition of these terms are as follows –

- **Wireless** – Lacking or not requiring a wire or wires pertaining to radio transmission.
- **Application** – A computer program or piece of computer software that is designed to do a specific task.
- **Protocol** – A set of technical rules about how information should be transmitted and received using computers.

■ WAP-WML Syntax

The topmost layer in the WAP architecture is made up of WAE (Wireless Application Environment), which consists of WML and WML scripting language.

WML scripting language is used to design applications that are sent over wireless devices such as mobile phones. This language takes care of the small screen and the low bandwidth of transmission. WML is an application of XML, which is defined in a document-type definition.

WML pages are called decks. They are constructed as a set of cards, related to each other with links. When a WML page is accessed from a mobile phone, all the cards in the page are downloaded from the WAP server to mobile phone showing the content.

WML commands and syntaxes are used to show content and to navigate between the cards. Developers can use these commands to declare variables, format text, and show images on the mobile phone.

■ WAP Program Structure

A WML program is typically divided into two parts – the **document prolog** and the **body**. Consider the following code –

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2/EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
  <card>

  ...
  </card>
  ...more cards...
</wml>
```

The first line of this text says that this is an XML document and the version is 1.0. The second line selects the document type and gives the URL of the **document type definition** (DTD). This DTD gives the full XML definition of WML. The DTD referenced is defined in WAP 1.1, but this header changes with the versions of the WML. The header must be copied exactly so that the tool kits automatically generate this prolog.

The body is enclosed within a <wml>...</wml> tag pair as shown above. The body of a WML document can consist of one or more of the following –

- Deck
- Card
- Content to be shown
- Navigation instructions

WML Commands

The commands used in WML are summarized as follows –

Formatting

Command	Description
<p>	Paragraph
	Bold
<big>	Large
	Emphasized
<i>	Italicized
<small>	Small
	Strongly Emphasized
<u>	Underlined
 	Line Break

Inserting Images

```

```

Using Tables

Command	Description
<table>	Definition of a table
<tr>	Defining a row
<td>	Defining a column
<Thead>	Table header

Variables

Declared as –

```
<setvar name="x" value="xyz"/>
```

Used as –

```
$ identifier or
$ (identifier) or
$ (Identifier; conversion)
```

Forms

Command	Description
<select>	Define single or multiple list
<input>	Input from user
<option>	Defines an option in a selectable list
<fieldset>	Defines a set of input fields
<optgroup>	Defines an option group in a selectable list

Task Elements

Command	Description
<go>	Represents the action of switching to a new card
<noop>	Says that nothing should be done
<prev>	Represents the action of going back to the previous card
<refresh>	Refreshes some specified card variables.

Events

The various events are as follows –

Command	Description
<do>	Defines a do event handler
<onevent>	Defines an onevent event handler
<postfield>	Defines a postfield event handler
<ontimer>	Defines an ontimer event handler
<onenterforward>	Defines an onenterforward handler

<code><onenterbackward></code>	Defines an onenterbackward handler
<code><onpick></code>	Defines an onpick event handler

Experiment No. 1

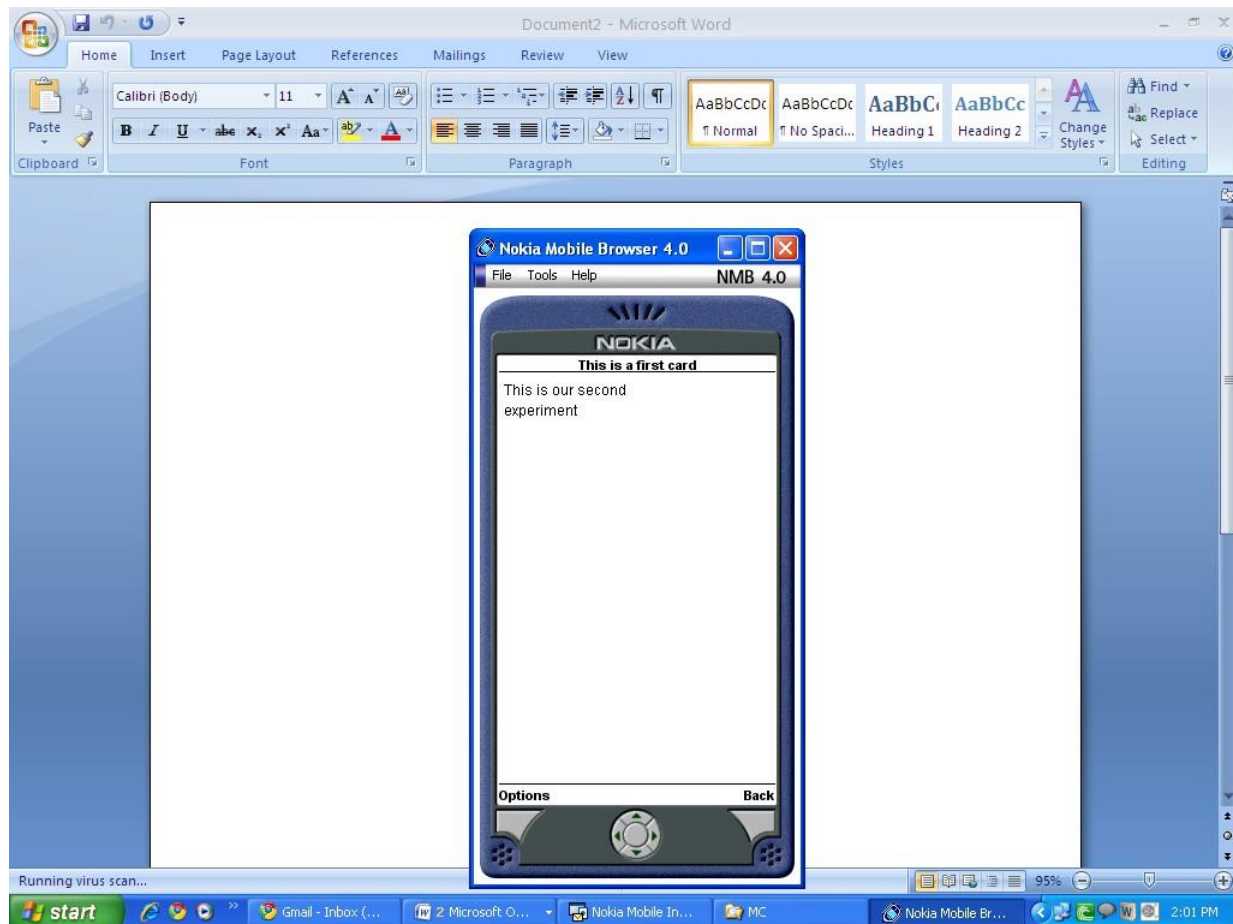
Aim: - Write a Program to create line break in WML

Requirements: - Nokia Mobile Internet toolkit

Code

```
<? Xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <card id="MainCard" title="This is a first card">
<p>This is our second <br/> experiment</p>
</card>
</wml>
```

Output:



Experiment No-2

AIM: Write a program in WML to illustrate the font size.

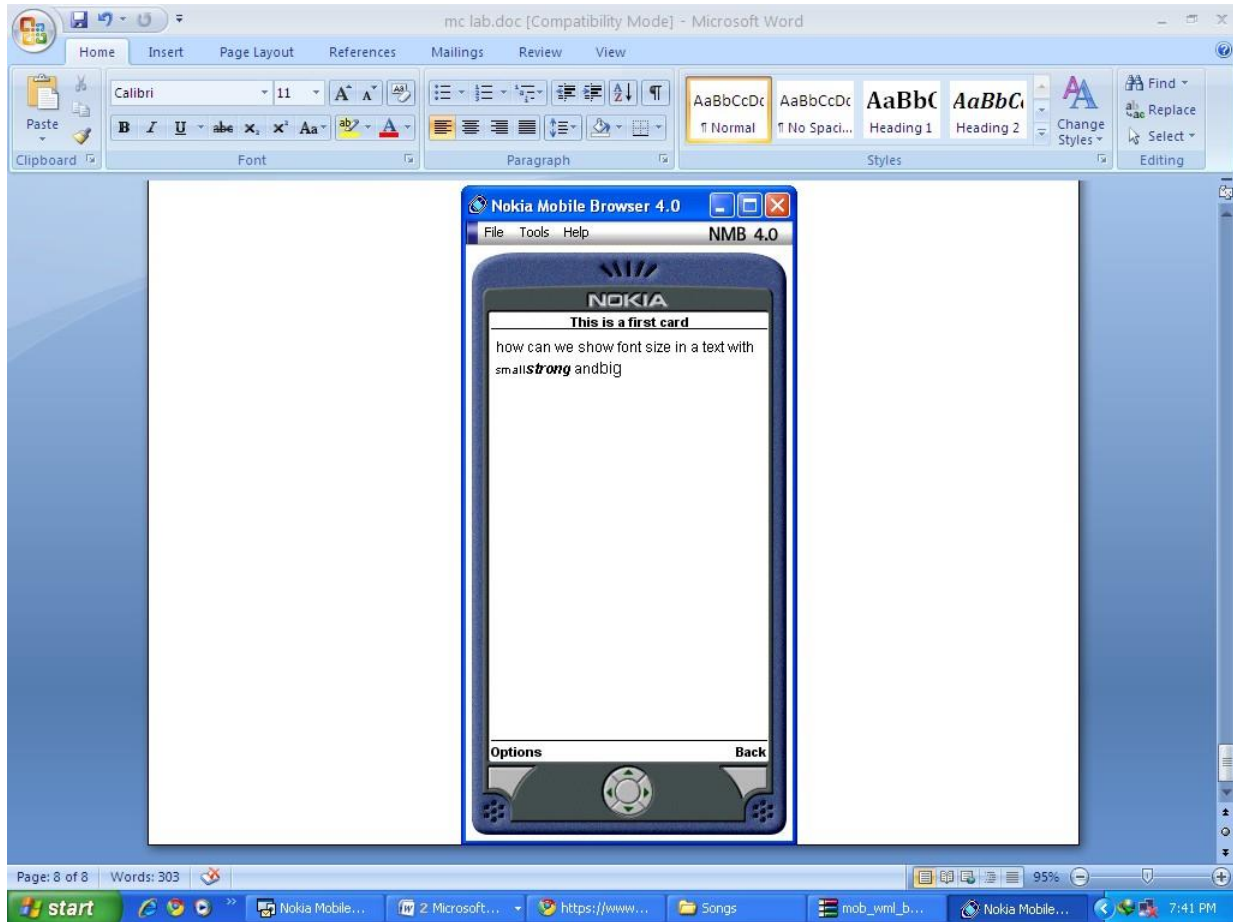
Requirements: - Nokia Mobile internet Toolkit.

Code

```
<? xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>

    <card id="MainCard" title="This is a first card">
    <p>how can we show font size in a text with
    <small>small</small>
    <i><strong>strong </strong></i>and<big>big</big>
    </p>
    </card>
</wml>
```


Output:



Experiment No. 3

Aim:- Program to make a table in the WML card.

Requirements:- Nokia Mobile internet Toolkit.

Code:

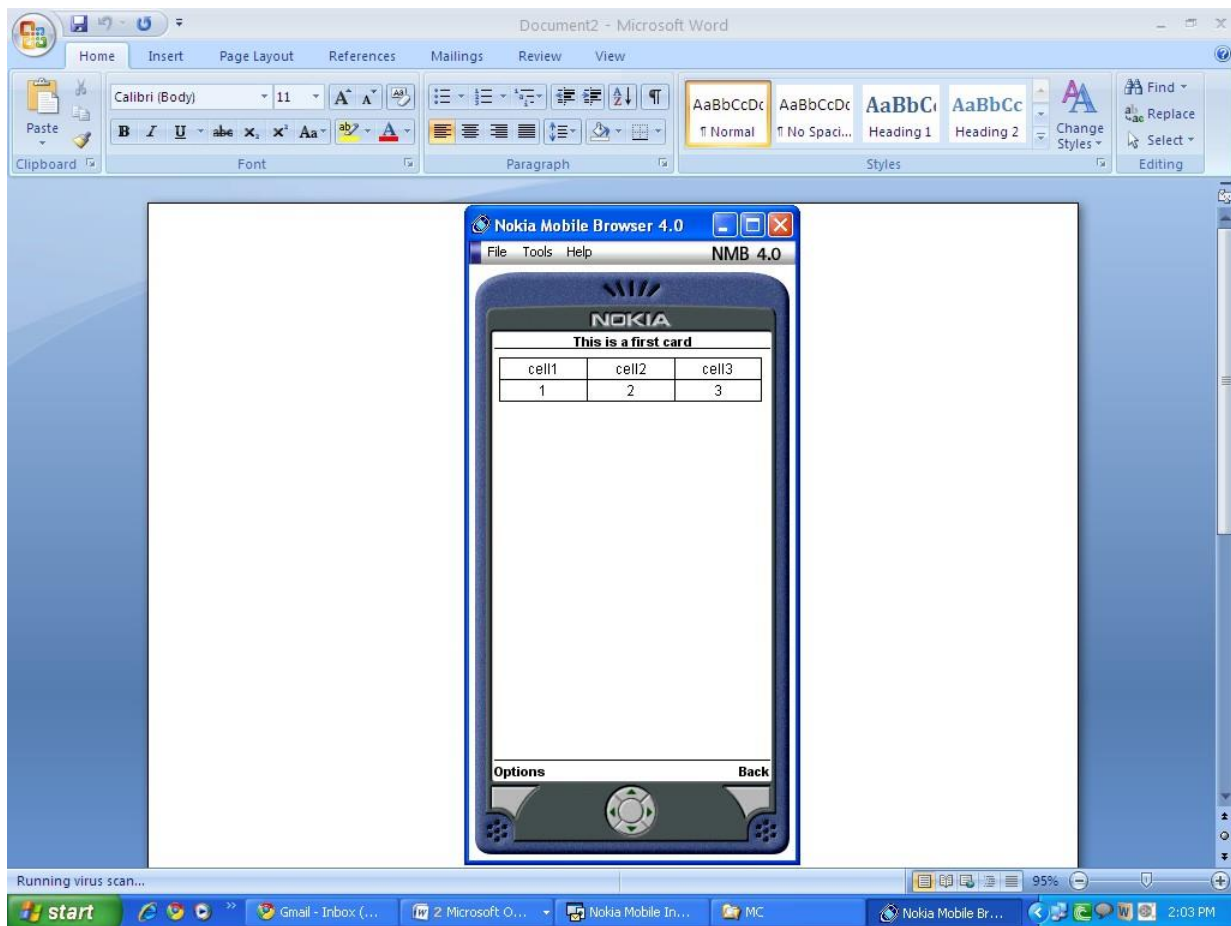
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<!-- THIS IS THE FIRST CARD IN THE DECK -->
<card id="pooja" title="This is a first card">
<p align="center">
<table columns="3">
<tr>
<td>cell1</td>
<td>cell2</td>
<td>cell3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</table>
```

</p>

</card>

</wml>

Output:



Experiment No-4

AIM: Write a program with two cards one for user input and other for displaying the result.

Requirements:- Nokia Mobile internet Toolkit.

CODE:

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">

<wml>

    <card id="CARD1" title="TUTORIAL">

        <p>

            <do type="ACCEPT" label="MY_SUBJECT">

                <go href="#CARD2"/>

            </do>

            <select name="NAME">

                <option value="Hindi">HINDI </option>

                <option value="English">ENGLISH</option>

                <option value="Maths">MATHS</option>

            </select>

        </p>

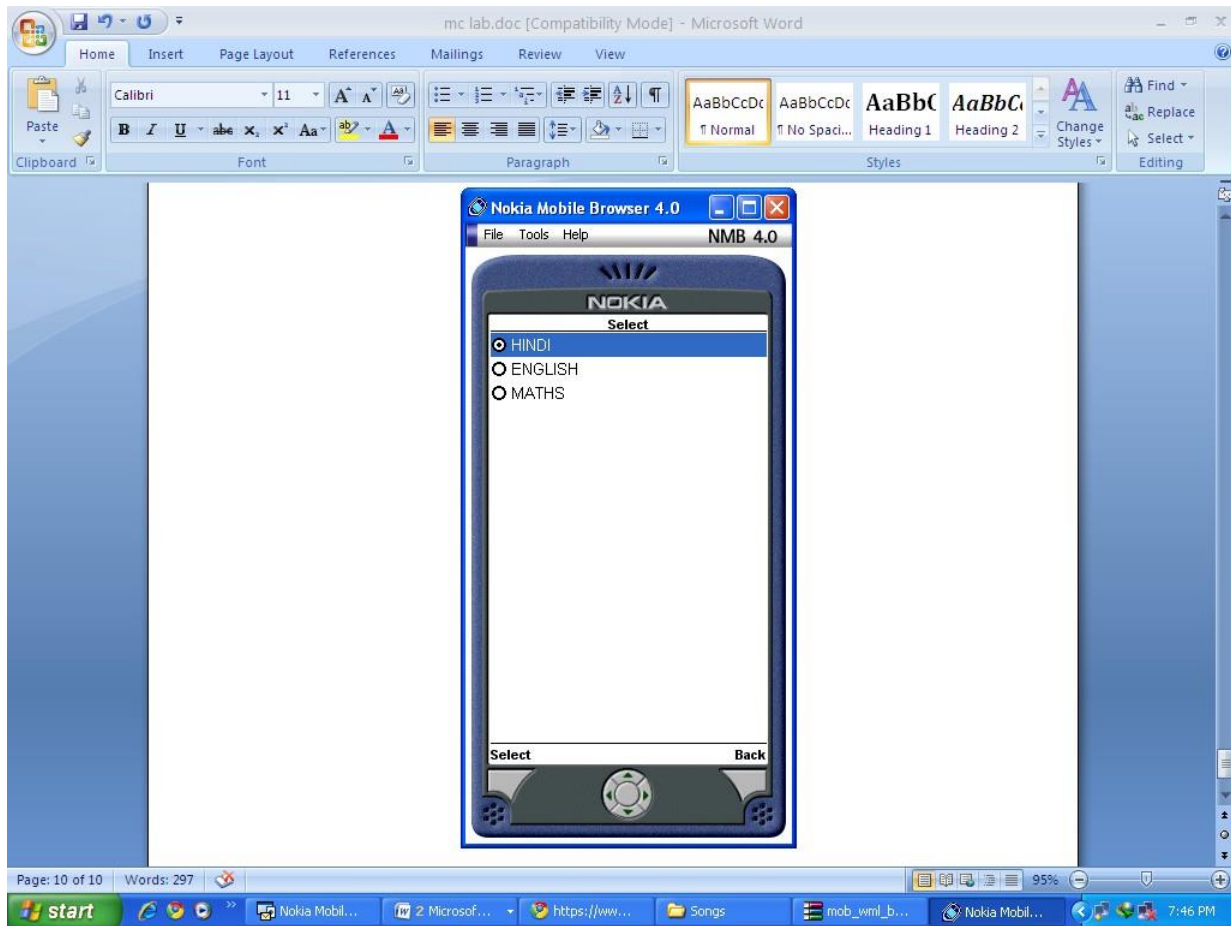
    </card>

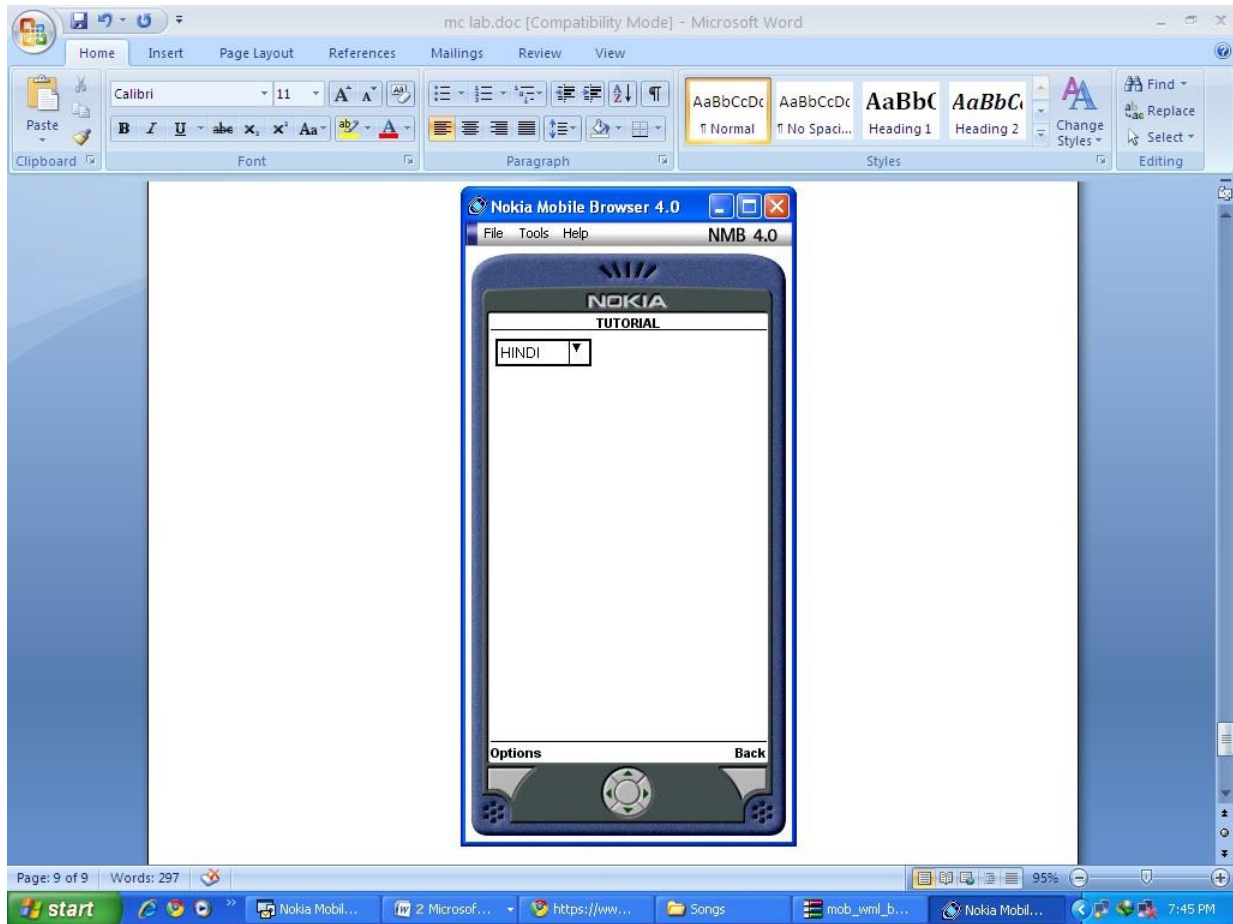
    <card id="CARD2" title="MY_SUBJECT">

        <p>YOU SELECTED: $(NAME)</p>

    </card></wml>
```

Output:





Experiment No. 5

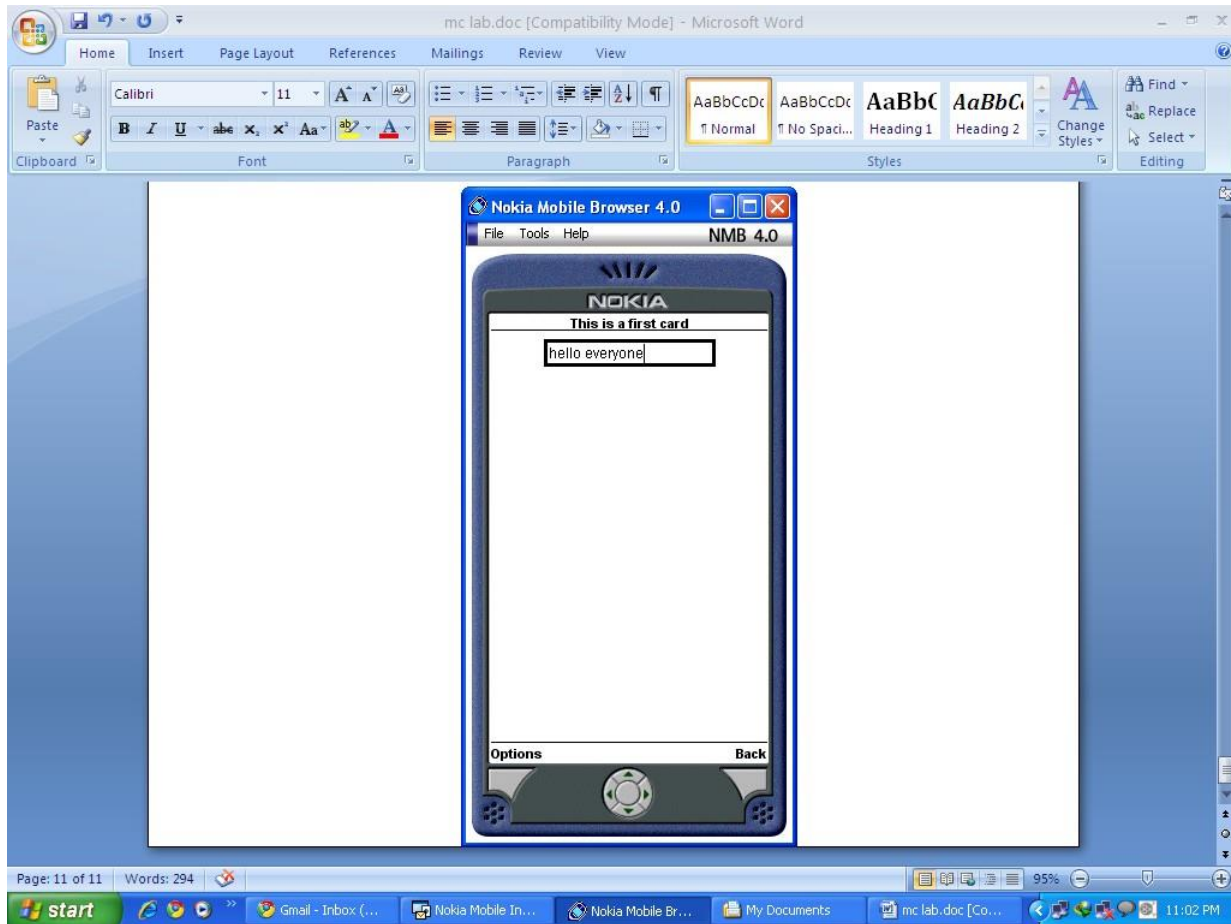
Aim:- Write a Program to introduce input box in a WML form.

Requirements:- Nokia Mobile internet Toolkit.

Code

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <!-- THIS IS THE FIRST CARD IN THE DECK -->
    <card id="ankit" title="This is a first card">
        <p align="center">
            Name:<input type="text" name="Name"
                size="15"/>
        </p>
    </card>
</wml>
```

Output:



Experiment No. 6

Aim:- Write a Program to a variable and its value in WML form.

Requirements:- Nokia Mobile internet Toolkit.

Code:

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.3//EN"

"http://www.wapforum.org/DTD/wml13.dtd">

<wml>

  <card id="CARD1" title="TUTORIAL">

    <p>

      <select name="NAME">

        <option value="WML">WML </option>

        <option value="HTML">HTML</option>

        <option value="Xml">XML</option>

      </select>

      <do type="ACCEPT" label="MY_SUBJECT">

        <go href="#CARD1"/>

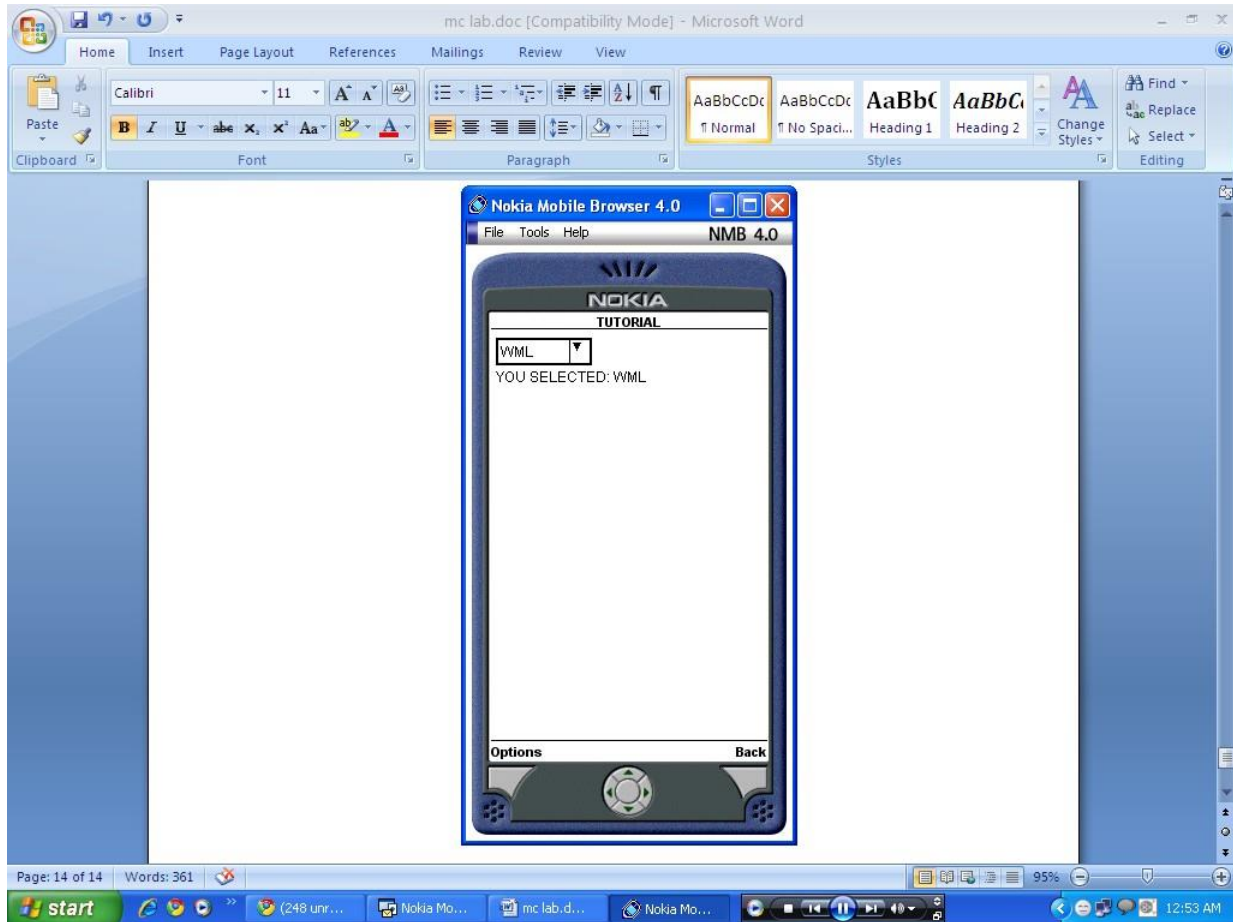
      </do>

      YOU SELECTED: $(NAME)</p>

    </card>

  </wml>
```

Output:



Experiment No. 7

Aim:- Write a program to perform navigation between WML cards (forward and backward both)

Requirements:- Nokia Mobile internet Toolkit.

Code:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <card id="c1" title="Card #1">
        <p align="center">
            <big><b>First Card</b></big><br/>
            Hello Everyone<br/>
            <a href="#c2">Card2</a><br/>
            <a href="#c3">Card3</a><br/>
        </p>
    </card>
    <card id="c2" title="Card #2">
        <p align="center">
            <big><b>Second
            Card</b></big><br/> Welcome to
            WML<br/>
            <a href="#c1">Back</a><br/>
            <a href="#c3">Next</a><br/>
        </p>
```

</card>

<card id="c3" title="Card #3">

<p align="center">

<big>Third Card</big>

Have a good day

 Back to Card1

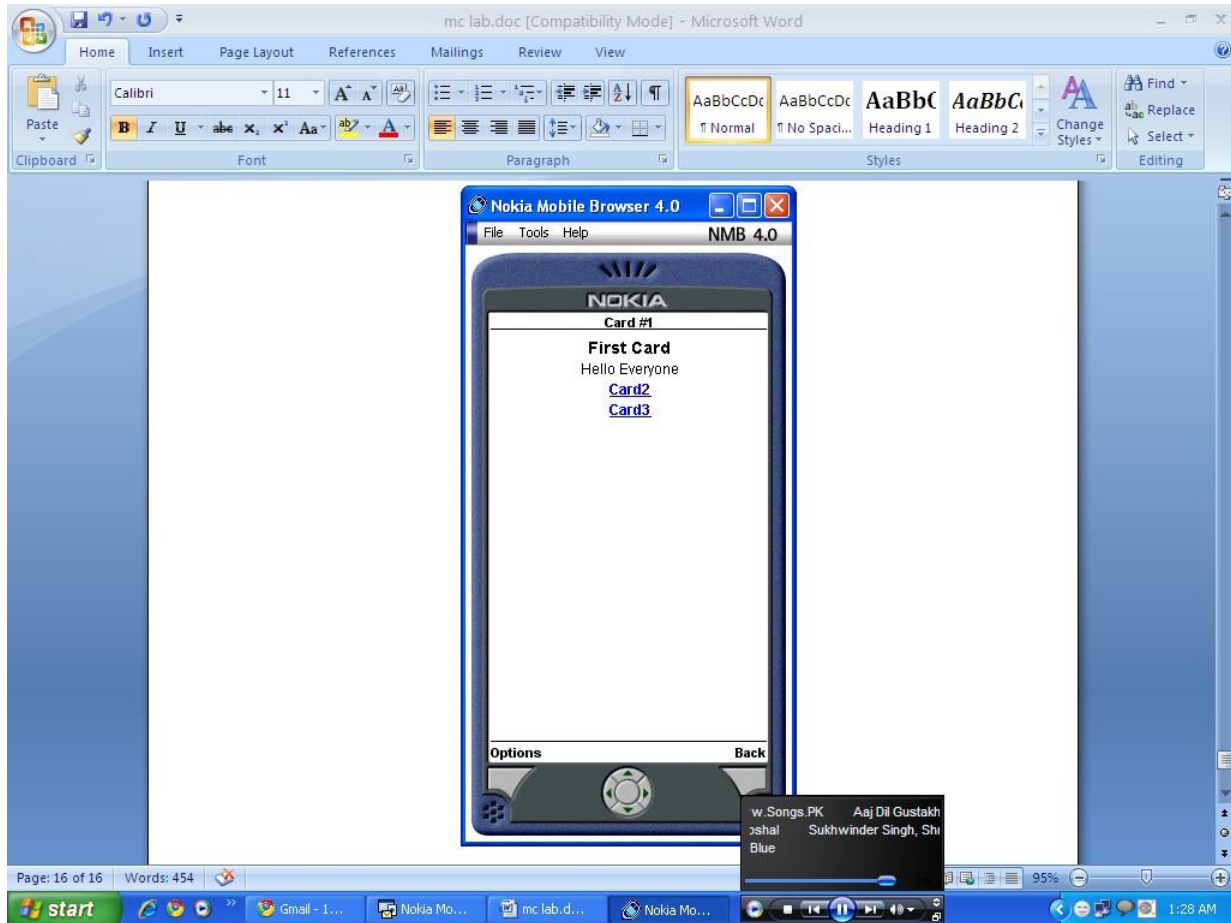
 Back to Card2

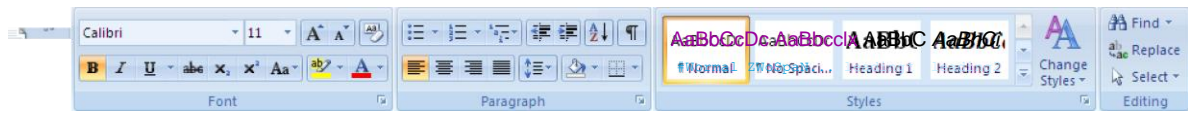
</p>

</card>

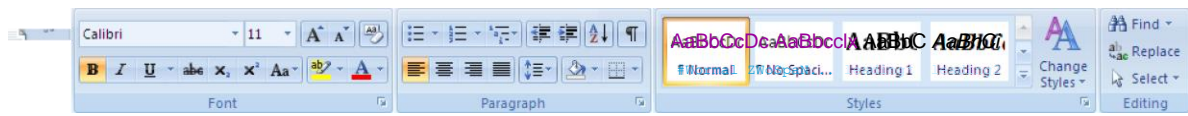
</wml>

Output:





Second Card
be/come to WML
[Back](#)
[Next](#)



Third Card
Have a good day
[Back to Card1](#)
[Back to Card2](#)

Experiment No. 8

Aim:- Write a Program to activate the current card in WML

Requirements:- Nokia Mobile Internet toolkit

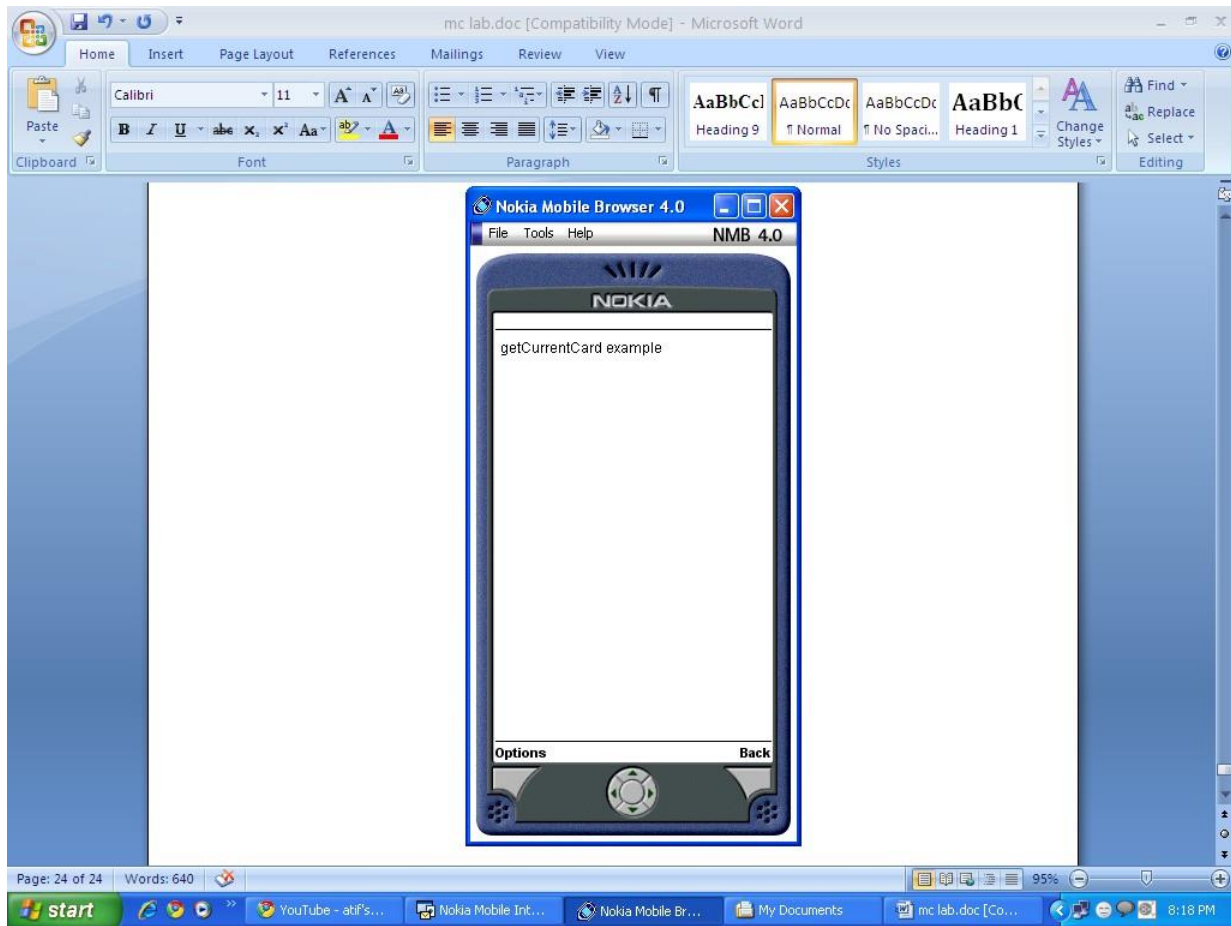
Code

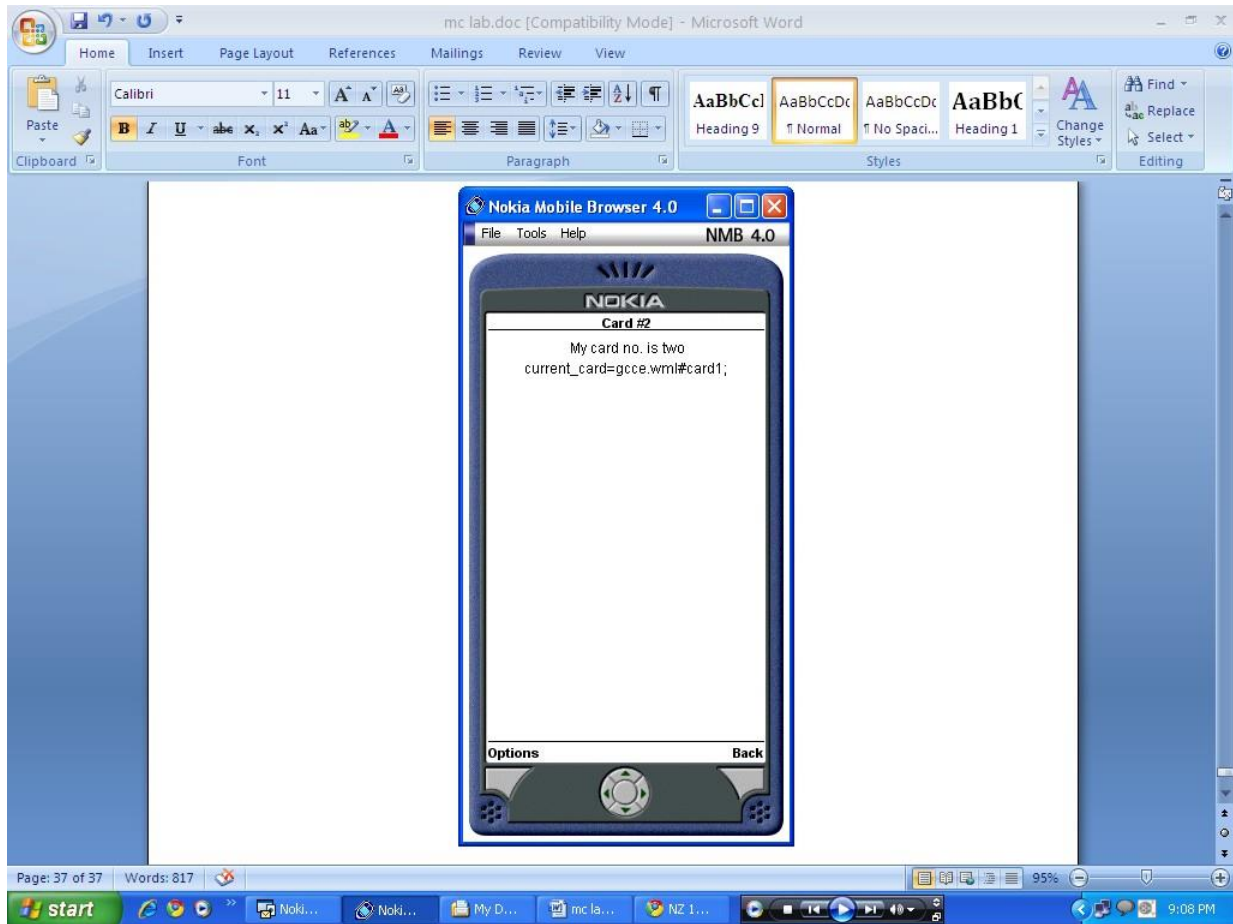
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <card id="card1" title="Card #1">
        <p>
            get current example</p>
        <do type="accept">
            <go href="GetCurrentCardEg.wmls#find()"/>
        </do>
    </card>
    <card id="card2" title="Card #2">
        <p align="center">
            My card no. is two
        </p>
    </card>
</wml>
```


Script File:

```
extern function find() {  
    var curcard=WMLBrowser.getCurrentCard();  
    WMLBrowser.setVar("currentcard",curcard);  
    WMLBrowser.go("GCCE.wml#card2");  
}
```

Output:





EXPERIMENT-9

AIM: Write a program in wml to print a message using WML script.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

    <card id="card1" title="wmlscript, tutorial">

        <p>

            <a href="helloworldEg1.wmls#helloworld()">Run wml script</a>

            <br/>

            $(message)

        </p>

    </card>

</wml>
```

WML Script :

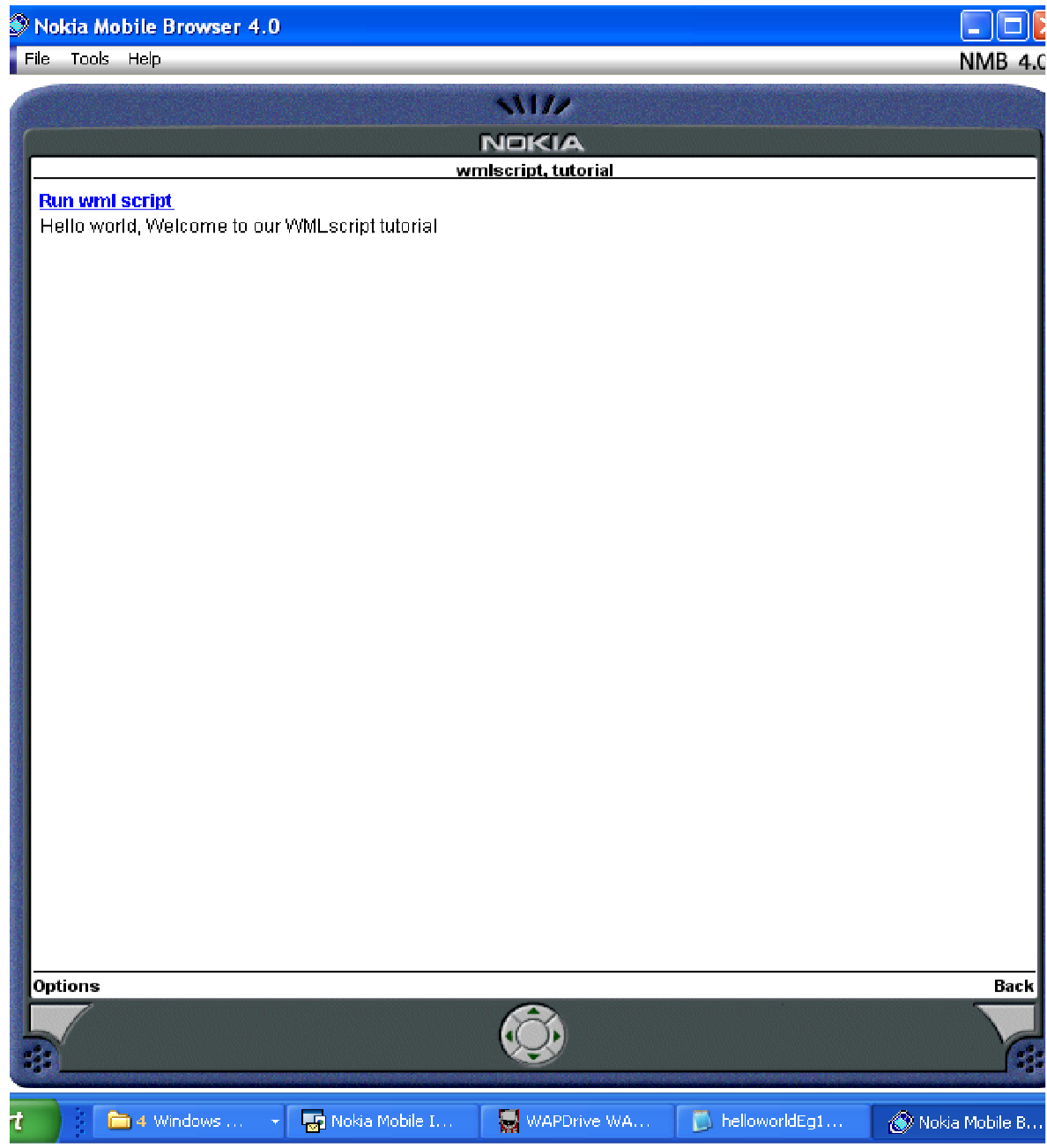
```
extern function helloworld()

{

WMLBrowser.setVar("message","Hello world Welcome to my wml
tutorial"); WMLBrowser.refresh();

}
```

OUTPUT :



EXPERIMENT-10

AIM: Write a program in wml to generate a random number using WML script.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

  <card id="card1" title="WMLScript Random Number">

    <p>

generate random number via WML Script

      <a href="random_number.wmls#RandomNumber()">Generate Number</a><br/>

        $(message)

        $(random)

$(message1)

    </p>

  </card>

</wml>
```

WML Script :

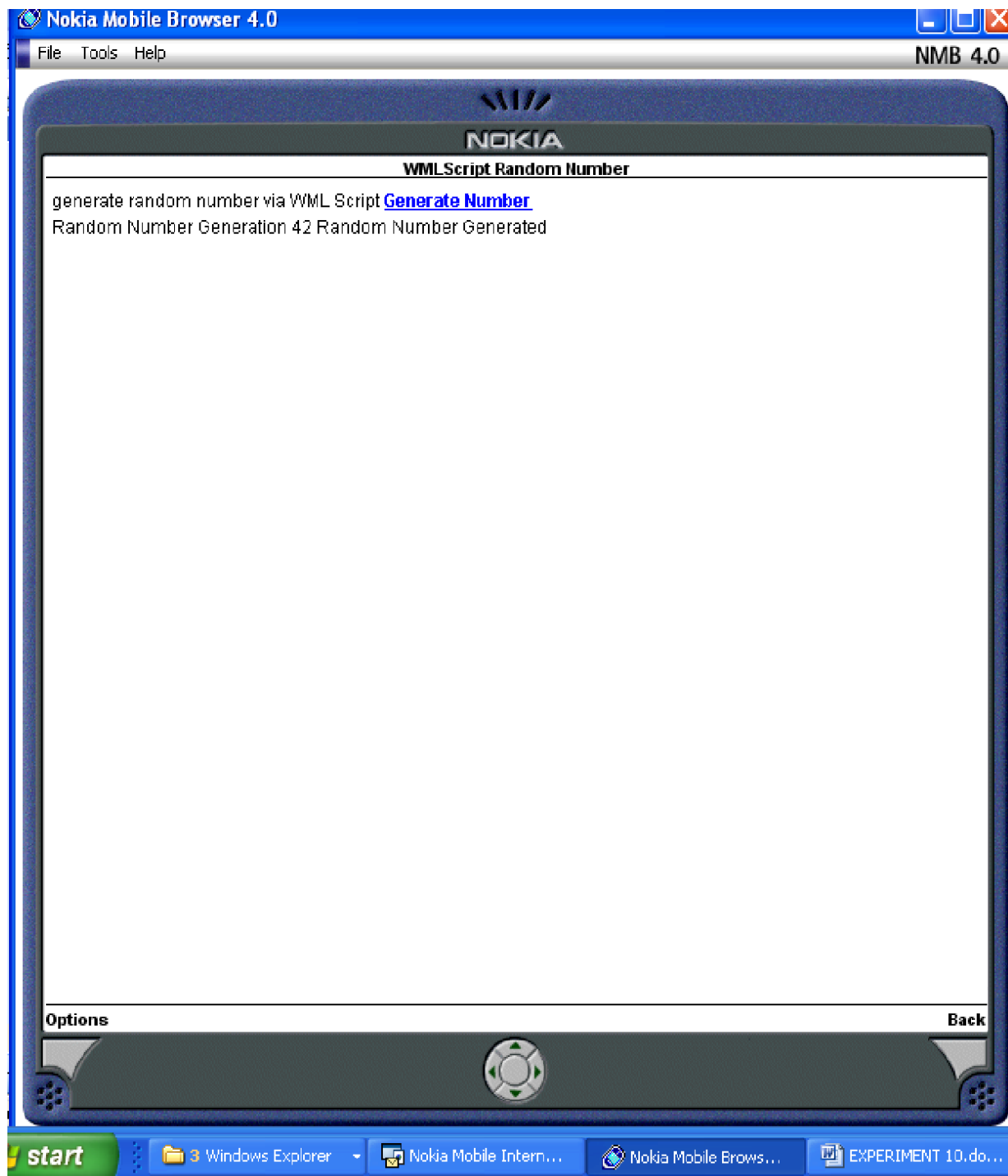
```
extern function RandomNumber()

{

WMLBrowser.setVar("message", "Random Number
Generation"); WMLBrowser.setVar("random",
Lang.random(100) ); WMLBrowser.setVar("message1",
"Random Number Generated"); WMLBrowser.refresh();

}
```

OUTPUT :



Experiment -11

Aim:- Write a Program to load a url by WML Script

Requirements:- Nokia Mobile Internet toolkit

Code

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <card id="card1" title="Card #1">
        <p>load string example</p>
        <do type="accept">
            <go href="isp.wmls#load()"/>
        </do>
    </card>
    <card id="card2">
        <p>
            url=$(urlstring);
        <br/>
        contenttype=$(content_type
    );
```


loadstring=\$(loadstr);

</p>

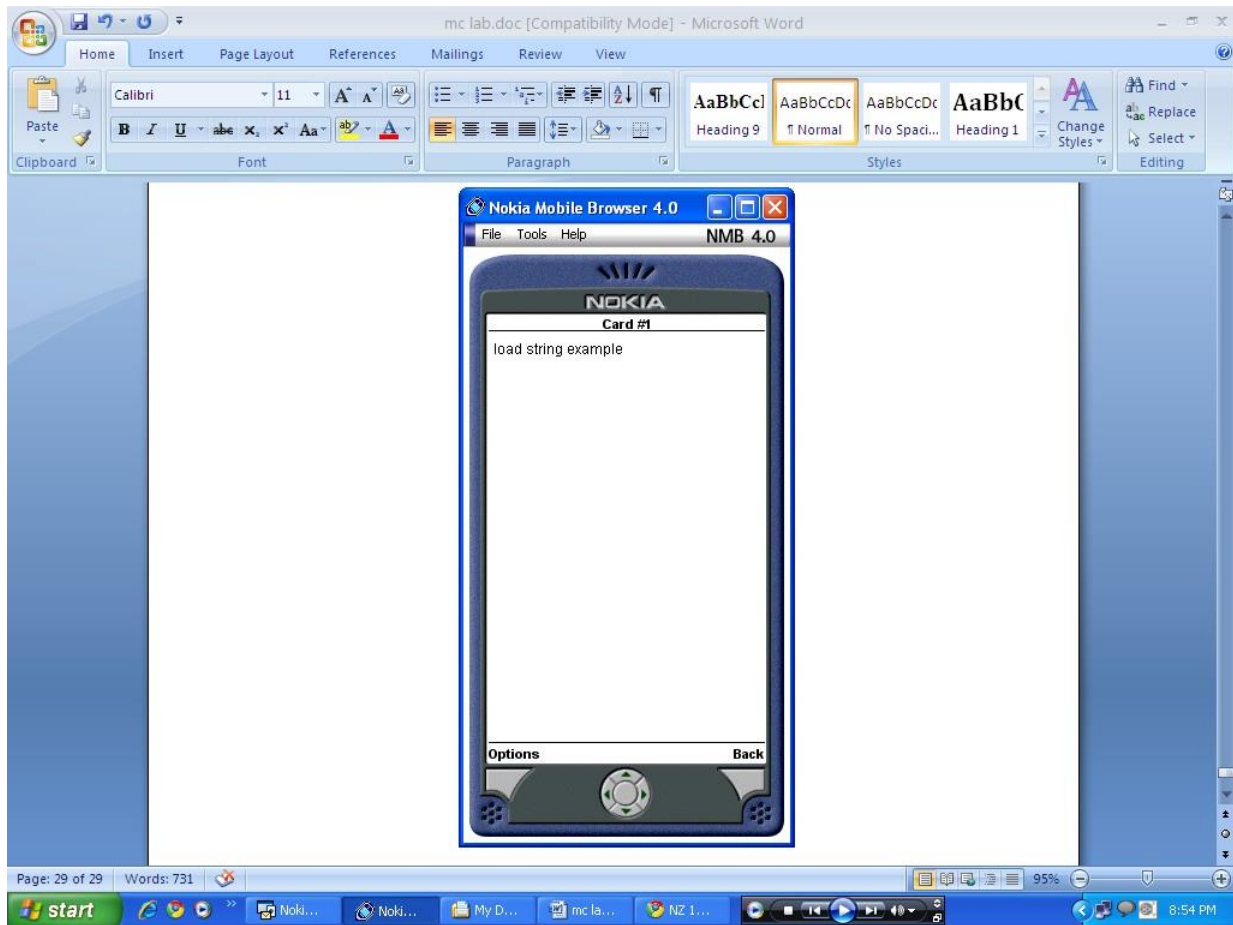
</card>

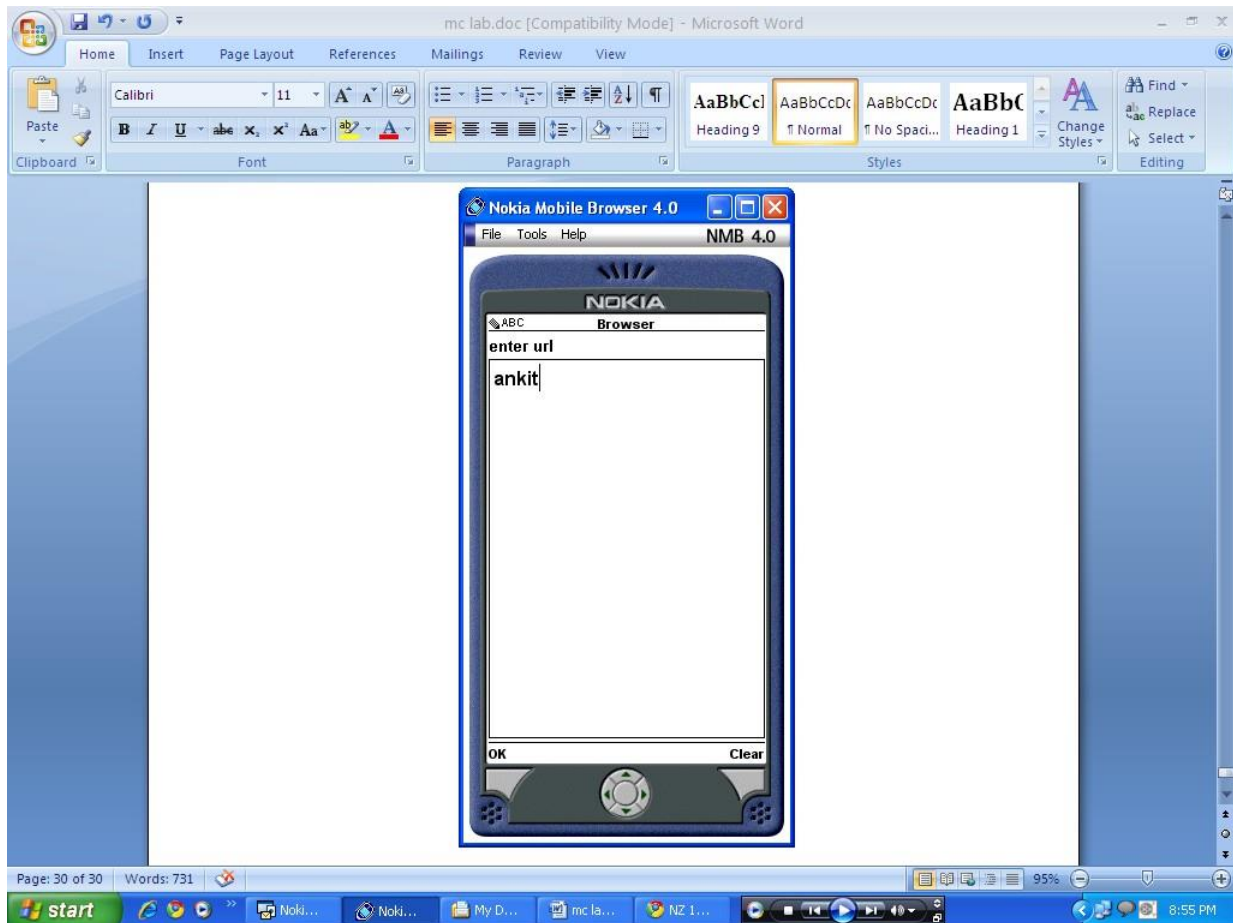
</wml>

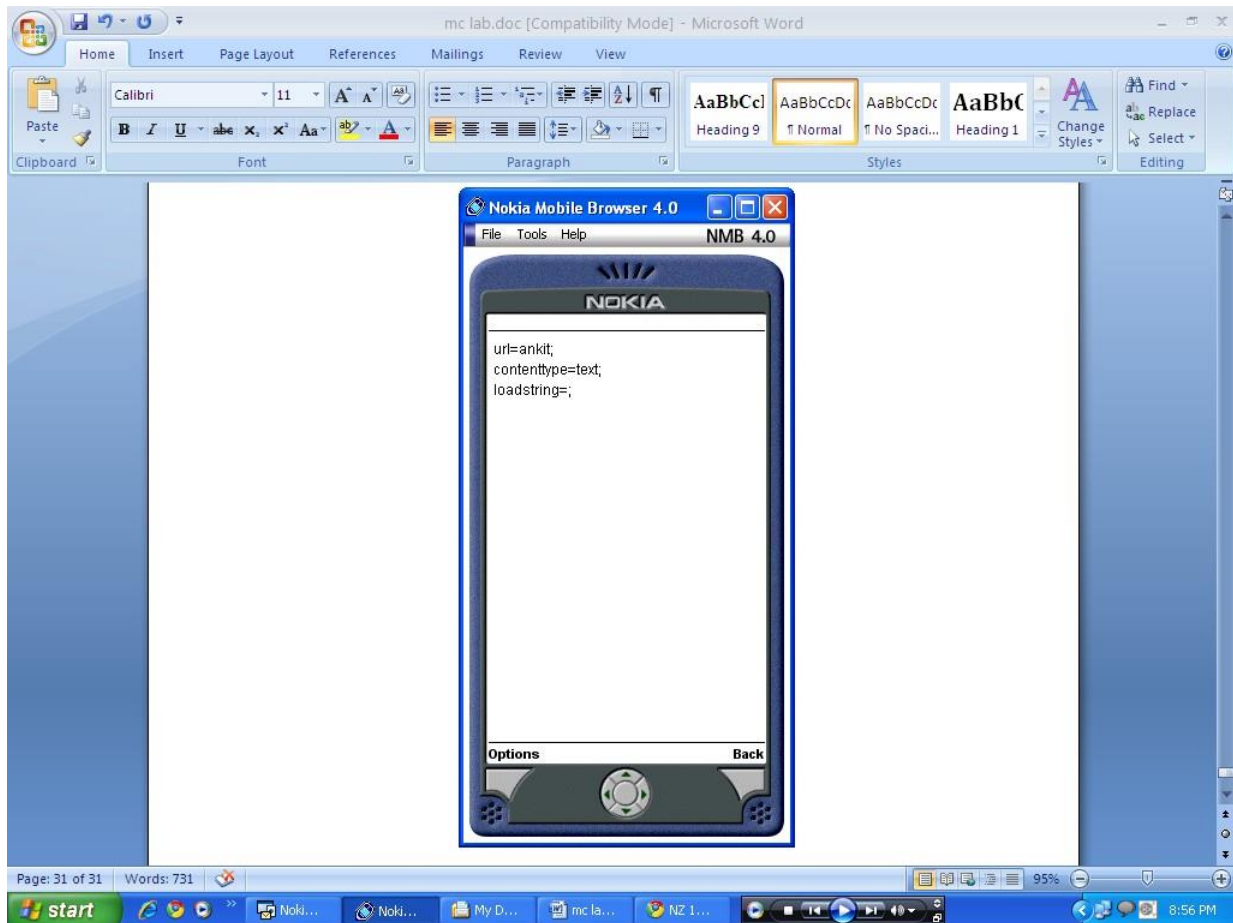
Script File:

```
extern function load() {  
    var urlstr=Dialogs.prompt("enter url", "");  
    var ctype=Dialogs.prompt("enter content  
type", "text"); var load=URL.loadString(urlstr,ctype);  
    WMLBrowser.setVar("urlstring",urlstr);  
    WMLBrowser.setVar("content_type",ctype);  
    WMLBrowser.setVar("loadstr",load);  
    WMLBrowser.go("laod.wml#card2");  
}
```

Output:







Experiment No. 12

Aim:- Write a Program to call a function by using WML Script to trim a screen.

Requirements:- Nokia Mobile Internet toolkit

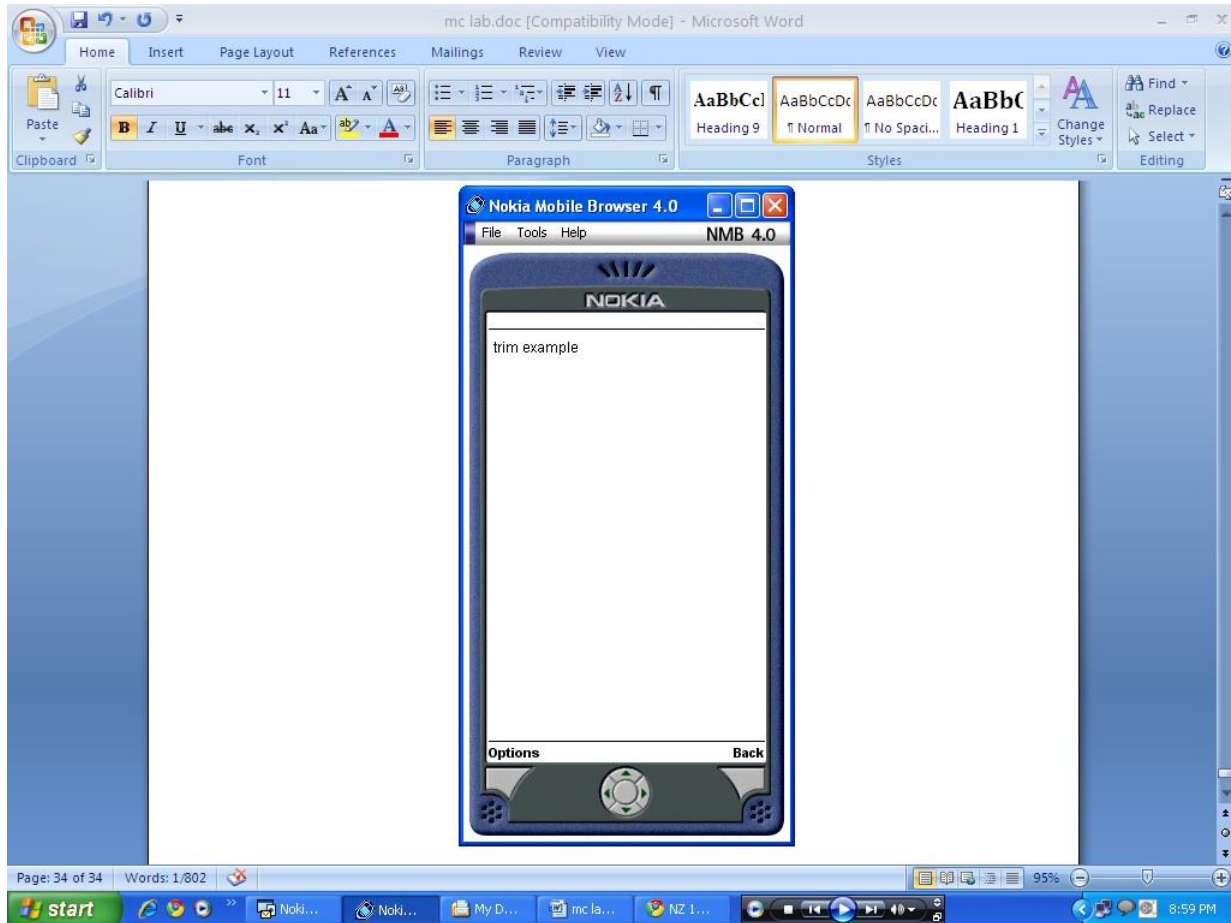
Code

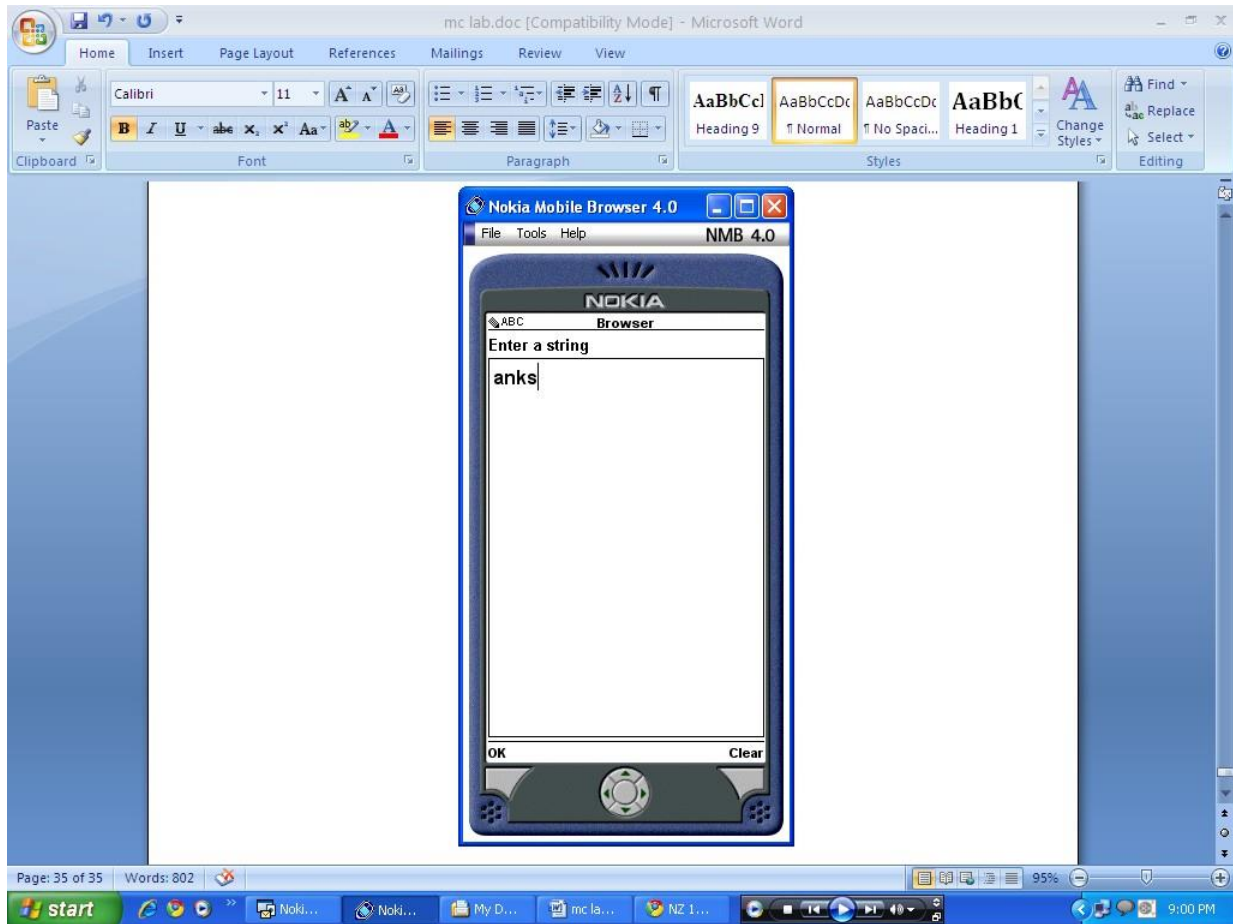
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <card id="c1">
        <p>
            trim example</p>
        <do type="accept">
            <go href="TrimExample.wmls#findtrim()"/>
        </do>
    </card>
    <card id="c2">
        <p>
            string=$(strn
            g)
        <br/>
            trim string=$(trimstrng)
        </p>
    </card>
</wml>
```

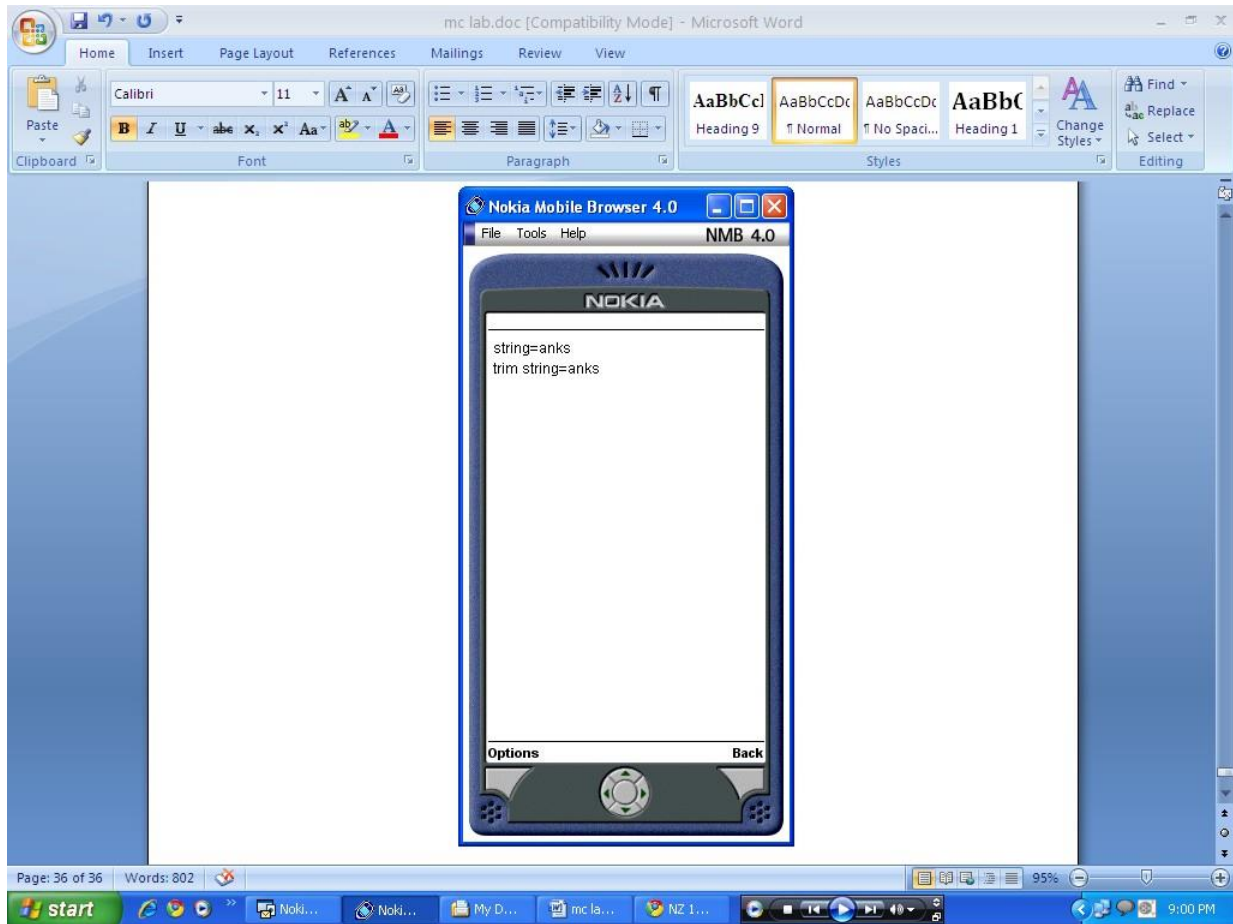
Script File:

```
extern function findtrim() {  
    var str= Dialogs.prompt("Enter a  
string",""); var trimstr=String.trim(str);  
    WMLBrowser.setVar("strng",str);  
    WMLBrowser.setVar("trimstrng",trimstr);  
    WMLBrowser.go("trm.wml#c2");  
}
```

Output:







EXPERIMENT-13

AIM: Write a program in wml to apply validation on a form using WML script.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

  <card id="card1" title="Registration Form">

    <p>

      <big>Registration Form</big><br/>

      Notice: Fields with * are
        required.<br/><br/> Form Validation
        Check via WML Script

      <b>$(errorMsg)</b><br/>

      * User name:<br/>

      <input name="username"/><br/>

      * Password (min. 8 characters):<br/>

      <input type="password" name="password"/><br/>

      * Email:<br/>

      <input
        name="email"/><br/>

      Name:<br/>

      <input name="name"/><br/>
```

Birthday (MMDDYYYY):

<input name="birthday" format="NNNNNNNN" emptyok="true"/>

Submit Form Data

</p>

</card>

</wml>

WML Script :

```
extern function validate()
```

```
{
```

```
    var form_username =
```

```
    String.trim(WMLBrowser.getVar("username")); var
```

```
    form_password = String.trim(WMLBrowser.getVar("password"));
```

```
    var form_email = String.trim(WMLBrowser.getVar("email"));
```

```
    var form_name = String.trim(WMLBrowser.getVar("name"));
```

```
    var form_birthday = String.trim(WMLBrowser.getVar("birthday"));
```

```
    if (" "==form_username){
```

```
        WMLBrowser.setVar("errorMsg", "The User Name field must not be  
empty."); WMLBrowser.refresh();
```

```
        return;
```

```
    }
```

```
    if (" "==form_password){
```

```
WMLBrowser.setVar("errorMsg", "The Password field must not be  
empty."); WMLBrowser.refresh();  
return;  
}
```

```
if (" "==form_email){  
WMLBrowser.setVar("errorMsg", "The Email field must not be  
empty."); WMLBrowser.refresh();  
return;  
}
```

```
if (String.length(form_password) < 8){  
WMLBrowser.setVar("errorMsg", "The password must contain at least 8  
characters since a short password is less secure.");  
WMLBrowser.refresh()  
; return;  
}
```

```
if (!isEmailValid(form_email)){  
WMLBrowser.setVar("errorMsg", "The email address's format is  
invalid."); WMLBrowser.refresh();  
return;  
}
```

```
if ("!="form_birthday && !isDateValid(form_birthday)){
```

```
WMLBrowser.setVar("errorMsg", "The date in the Birthday field is  
invalid."); WMLBrowser.refresh();  
return;  
}
```

```
submit_form(form_username, form_password, form_email, form_name, form_birthday);  
}
```

```
function isEmailValid(emailAddr)  
{  
  if (String.elements(emailAddr, "@") !=  
    2) return false;  
  var element_1 = String.elementAt(emailAddr, 0,  
    "@");      var      element_2      =  
    String.elementAt(emailAddr, 1, "@");  if  
    ("=="element_1 || ""==element_2)  
    return false;  
    return true;  
}
```

```
function isDateValid(date)  
{  
  var mm = String.subString(date, 0, 2);  
  var dd = String.subString(date, 2, 2);  
  var yyyy = String.subString(date, 4, 4);
```

```
mm =  
Lang.parseInt(mm); dd =  
Lang.parseInt(dd); yyyy  
= Lang.parseInt(yyyy);
```

```
if (mm<1 || mm>12)  
return false;
```

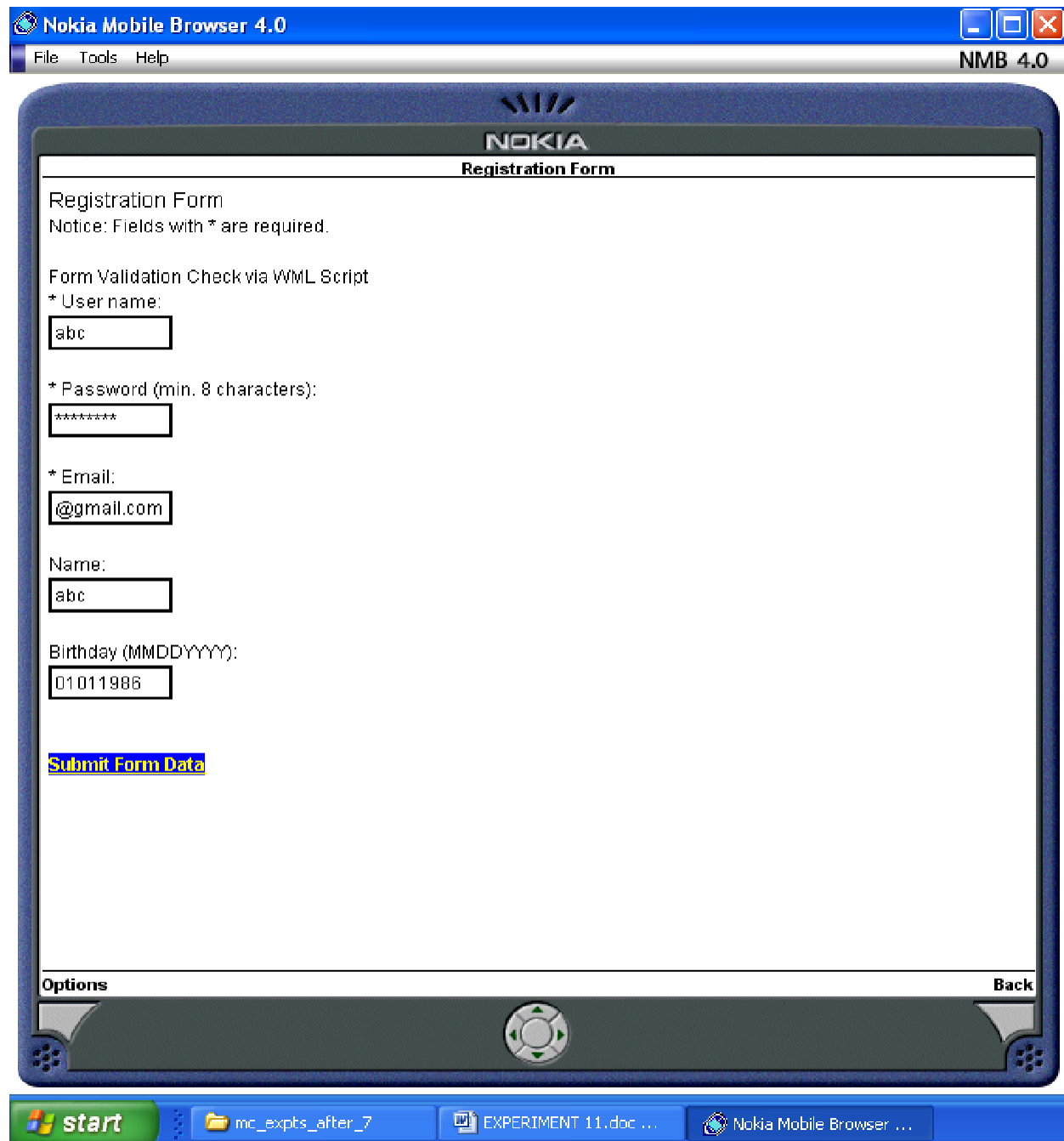
```
var maxDay = 31;  
if (4==mm || 6==mm || 9==mm || 11==mm)  
maxDay = 30;  
if (2==mm){  
if (0 == yyyy%4)  
maxDay = 29; // Leap  
year else  
maxDay = 28;  
}
```

```
if (dd<1 || dd>maxDay)  
return false;  
return true;  
}
```

```
function submit_form(form_username, form_password, form_email, form_name, form_birthday)  
{  
WMLBrowser.setVar("errorMsg", "");
```

```
WMLBrowser.setVar("username", form_username);  
WMLBrowser.setVar("password", form_password);  
WMLBrowser.setVar("email", form_email);  
WMLBrowser.setVar("name", form_name);  
WMLBrowser.setVar("birthday", form_birthday);  
  
WMLBrowser.go("validateFormEg1_success.wml");  
}
```

OUTPUT :



The screenshot shows a Nokia Mobile Browser 4.0 window. The browser's title bar is blue with the text "Nokia Mobile Browser 4.0" and standard window controls. Below the title bar is a menu bar with "File", "Tools", and "Help". The main content area is a registration form titled "Registration Form" with a Nokia logo at the top. The form includes a notice about required fields, a validation check via WML Script, and several input fields for user information. A "Submit Form Data" link is at the bottom of the form. The browser's status bar at the bottom shows a "start" button, a folder icon for "mc_expts_after_7", a document icon for "EXPERIMENT 11.doc ...", and a "Nokia Mobile Browser ..." button.

Nokia Mobile Browser 4.0
File Tools Help NMB 4.0

NOKIA
Registration Form

Registration Form
Notice: Fields with * are required.

Form Validation Check via WML Script

* User name:

* Password (min. 8 characters):

* Email:

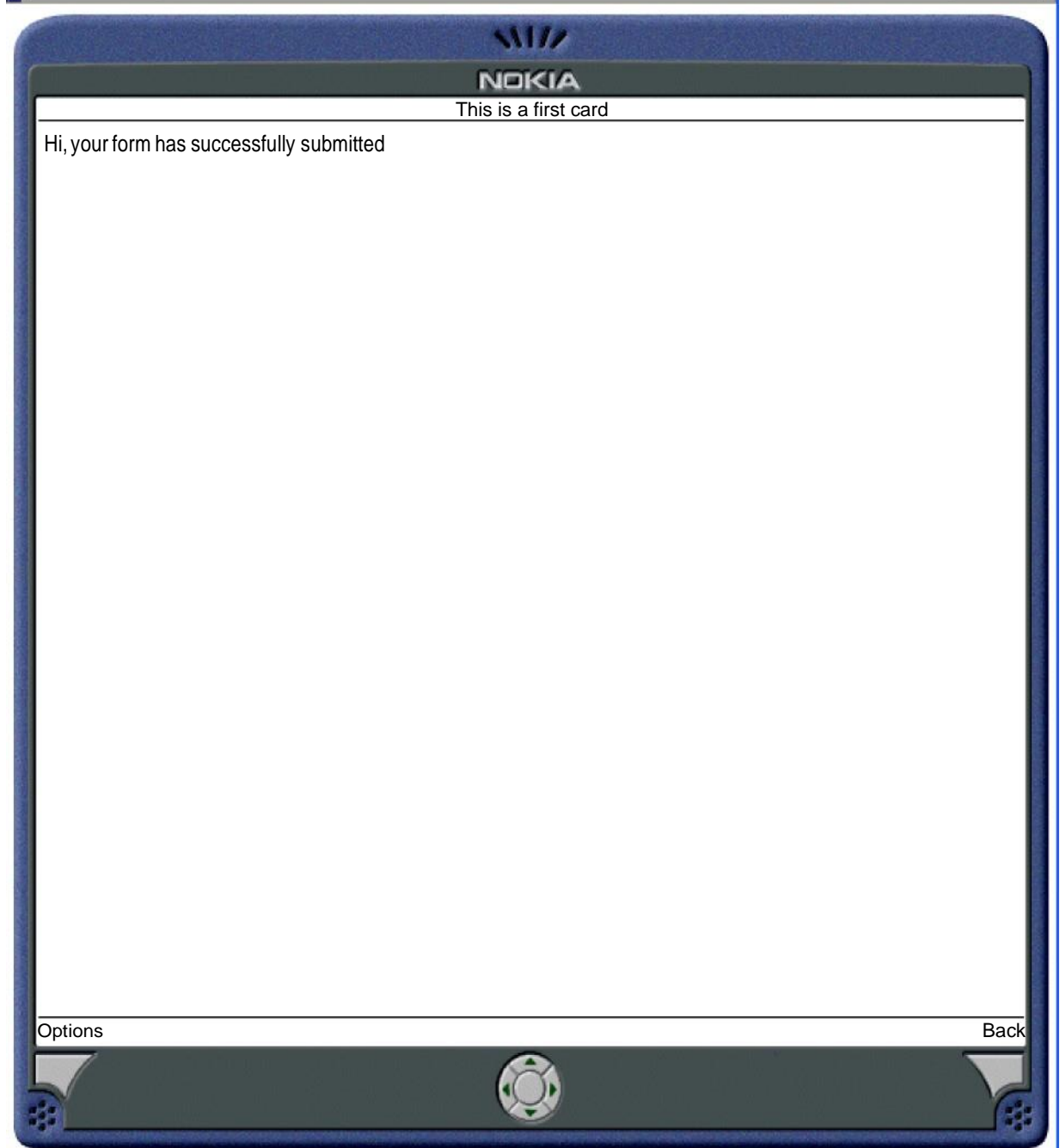
Name:

Birthday (MMDDYYYY):

[Submit Form Data](#)

Options Back

start mc_expts_after_7 EXPERIMENT 11.doc ... Nokia Mobile Browser ...



EXPERIMENT-14

AIM: Write a program in wml to convert date format using WML script.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

  <card id="card1" title="WMLScript Tutorial">

    <p>

      Please enter a date in the MM-DD-YYYY format:<br/>

      <input name="datef1"/><br/>

      <a href="passByRefEg1.wmls#changeDateFormat('${datef1}')">Run WMLScript</a><br/><br/>

    </p>

    <pre>$(result)</pre>

  </card>

</wml>
```

WML Script :

```

extern function changeDateFormat(date)
{
    WMLBrowser.setVar("datef2", date);

    parseDate("datef2", "day", "month", "year");

    var datef2 =
    WMLBrowser.getVar("datef2"); var day =
    WMLBrowser.getVar("day");
    var month =
    WMLBrowser.getVar("month"); var year =
    WMLBrowser.getVar("year");

    WMLBrowser.setVar("result", "Day: " + day + "\nMonth: " + month + "\nYear: " + year +
    "\nDate after conversion: " + datef2);

    WMLBrowser.refresh();
}

```

```

function parseDate(dateWMLVar, dayWMLVar, monthWMLVar, yearWMLVar)
{
    var date =
    WMLBrowser.getVar(dateWMLVar); var
    month = String.elementAt(date, 0, "-");
    var day = String.elementAt(date, 1, "-
    "); var year = String.elementAt(date,
    2, "-"); date = day + "/" + month + "/"
    + year;

    WMLBrowser.setVar(dateWMLVar, date);
    WMLBrowser.setVar(dayWMLVar, day);

```

```
WMLBrowser.setVar(monthWMLVar, month);  
WMLBrowser.setVar(yearWMLVar, year);  
}
```

OUTPUT :



EXPERIMENT-15

AIM: Write a program in wml to display an image.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>

  <card id="card1" title="Image in WML">

    <p>

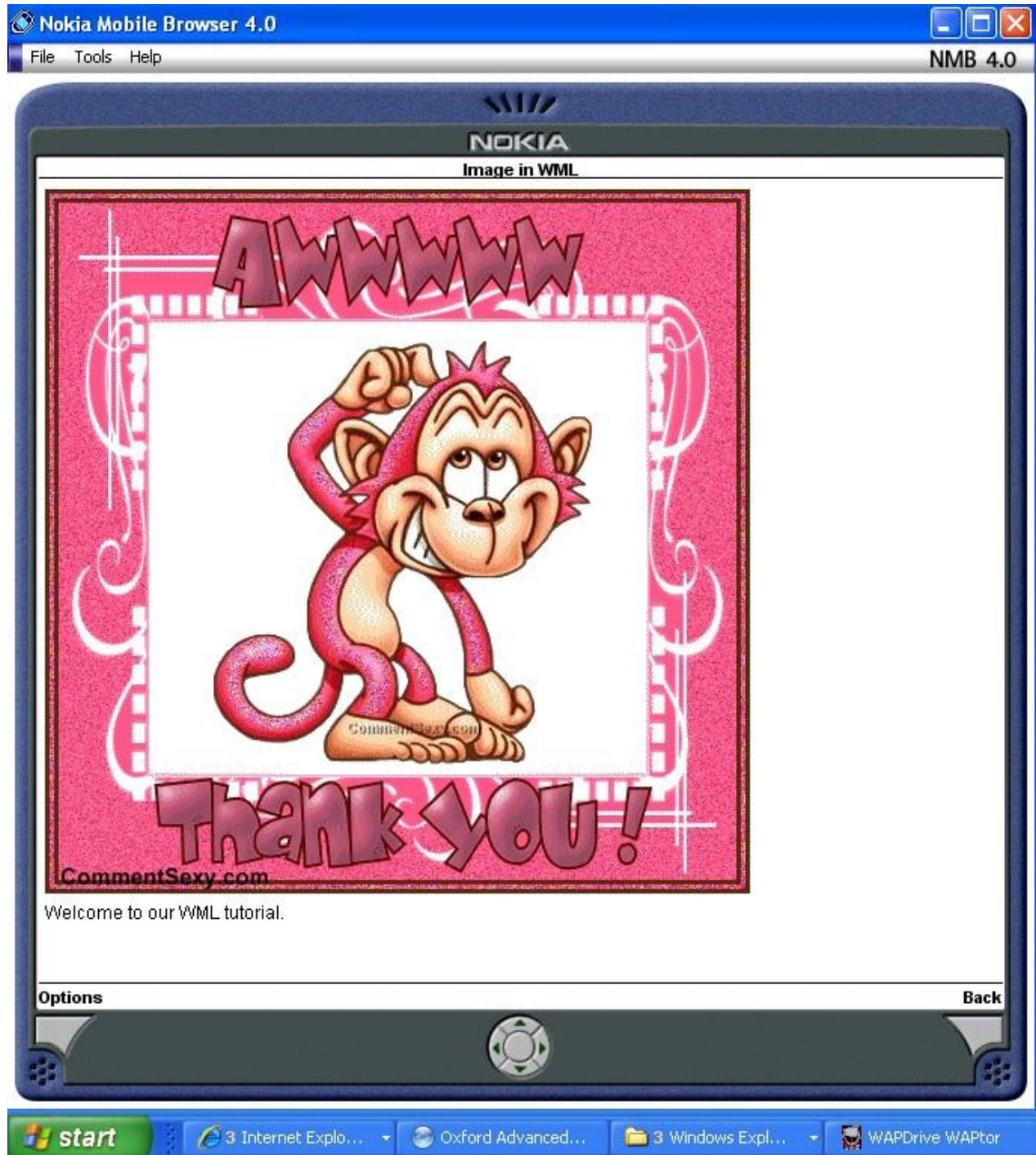
      <br/>
      Welcome to our WML tutorial.

    </p>

  </card>

</wml>
```

OUTPUT :



EXPERIMENT-16

AIM: Write a program in wml to apply timer tag for 3 sec and display another page after 3sec.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card ontimer="test.wml">

    <timer value="30"/>

    <p>Hello World!</p>

</card>

</wml>
```

test.wml

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<!-- created by EasyPad WAPtor (http://www.waptop.net/) -->

<wml>
```



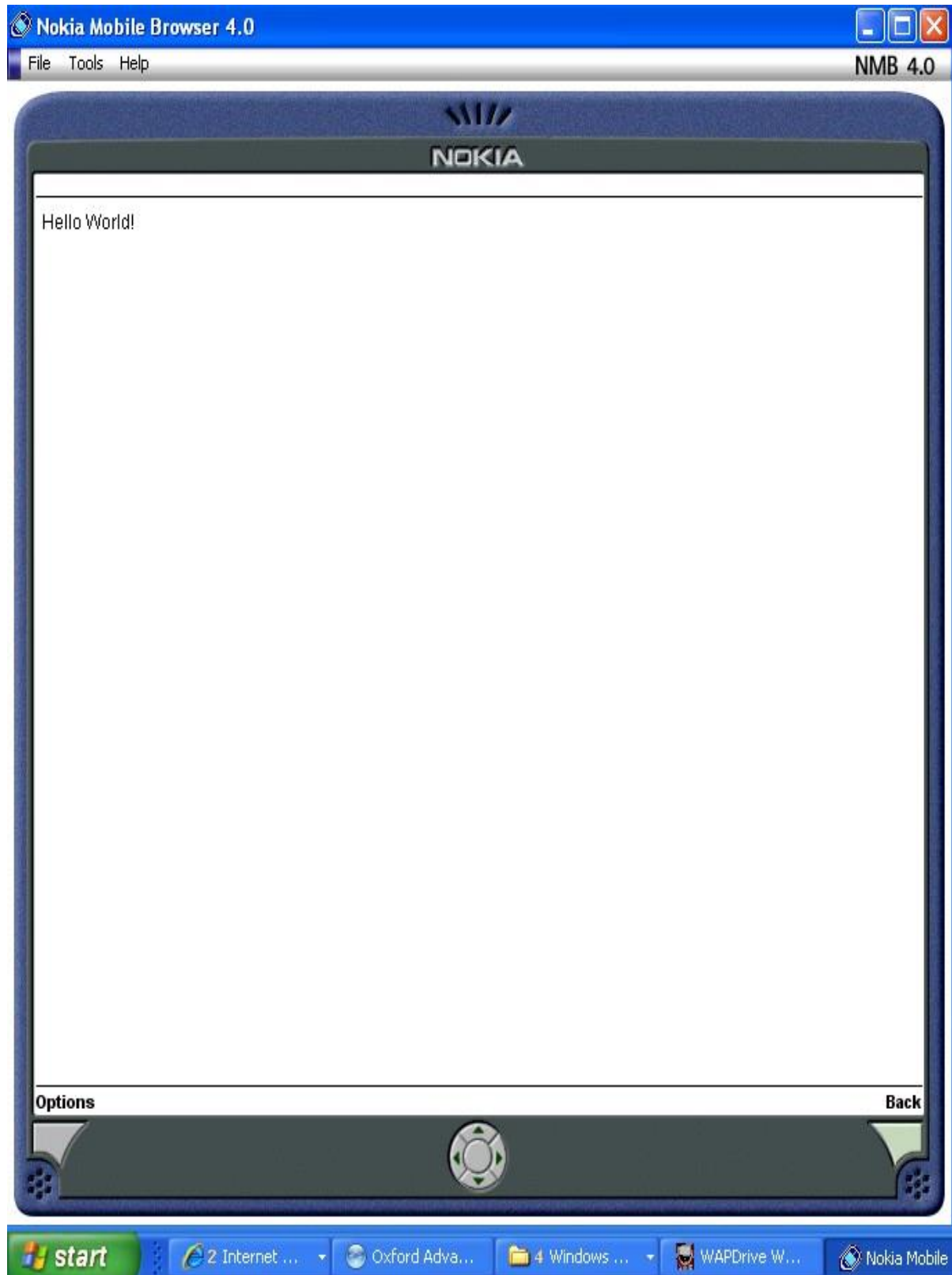
```
<card id="MainCard" title="This is a first card">
```

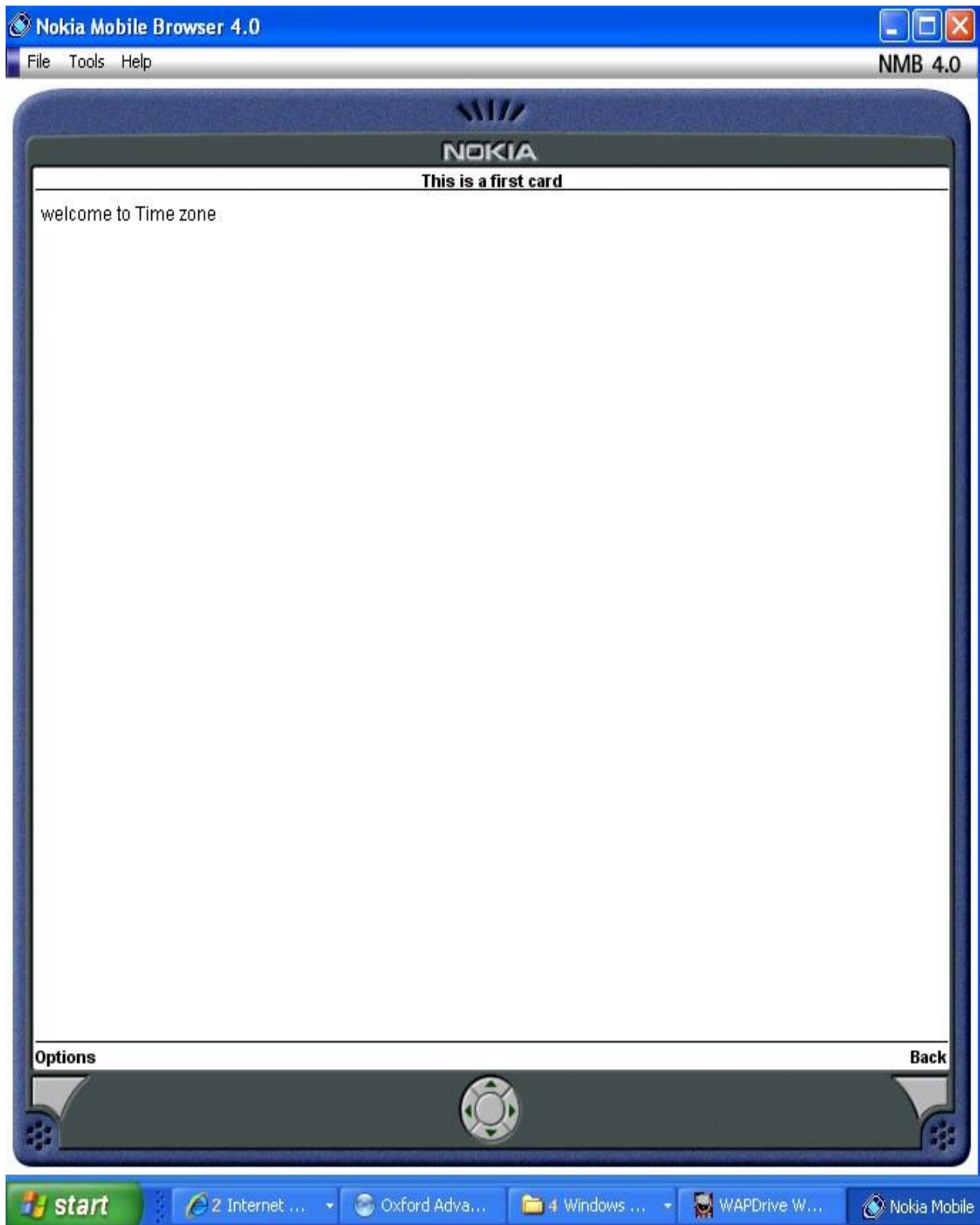
```
<p>welcome to Time zone</p>
```

```
</card>
```

```
</wml>
```

OUTPUT :





EXPERIMENT-17

AIM: Write a program to create a simple calculator using wml script in wml.

CODE:

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>

  <card id="card1" title="calculator">

    <p>

      <big>CALCULATOR</big><br/>

      Enter first number:<br/>

      <input name="first"/><br/>

      Enter second
      number:<br/>

      <input
      name="second"/><br/>

      result :<br/>

      <input name="result" value="$(result)" /><br/>

      <a href="calcu.wmls#add()">ADD</a><br/>

      <a href="calcu.wmls#sub()">SUBTRACTION</a><br/>

      <a href="calcu.wmls#multi()">MULTIPLY</a><br/>
```

```
<a href="calcu.wmls#divi()">DIVISION</a><br/>
```

```
</p>
```

```
</card>
```

```
</wml>
```

WML Script :

```
extern function add()
```

```
{
```

```
var first= WMLBrowser.getVar("first");
```

```
var second= WMLBrowser.getVar("second");
```

```
first= Lang.parseInt(first);
```

```
second = Lang.parseInt(second);
```

```
WMLBrowser.setVar("result", first + second);
```

```
WMLBrowser.refresh();
```

```
}
```

```
extern function sub()
```

```
{var first= WMLBrowser.getVar("first");
```

```
var second= WMLBrowser.getVar("second");
```

```
first= Lang.parseInt(first);
```

```
second = Lang.parseInt(second);
```

```
WMLBrowser.setVar("result", first -
```

```
second); WMLBrowser.refresh();
```

```
}
```

```
extern function multi()
```

```
{
```

```
var first= WMLBrowser.getVar("first");
```

```
var second= WMLBrowser.getVar("second");
```

```
first= Lang.parseInt(first);
```

```
second = Lang.parseInt(second);
```

```
WMLBrowser.setVar("result", first * second);
```

```
WMLBrowser.refresh();
```

```
}
```

```
extern function divi()
```

```
{
```

```
var first= WMLBrowser.getVar("first");
```

```
var second= WMLBrowser.getVar("second");
```

```
first= Lang.parseInt(first);
```

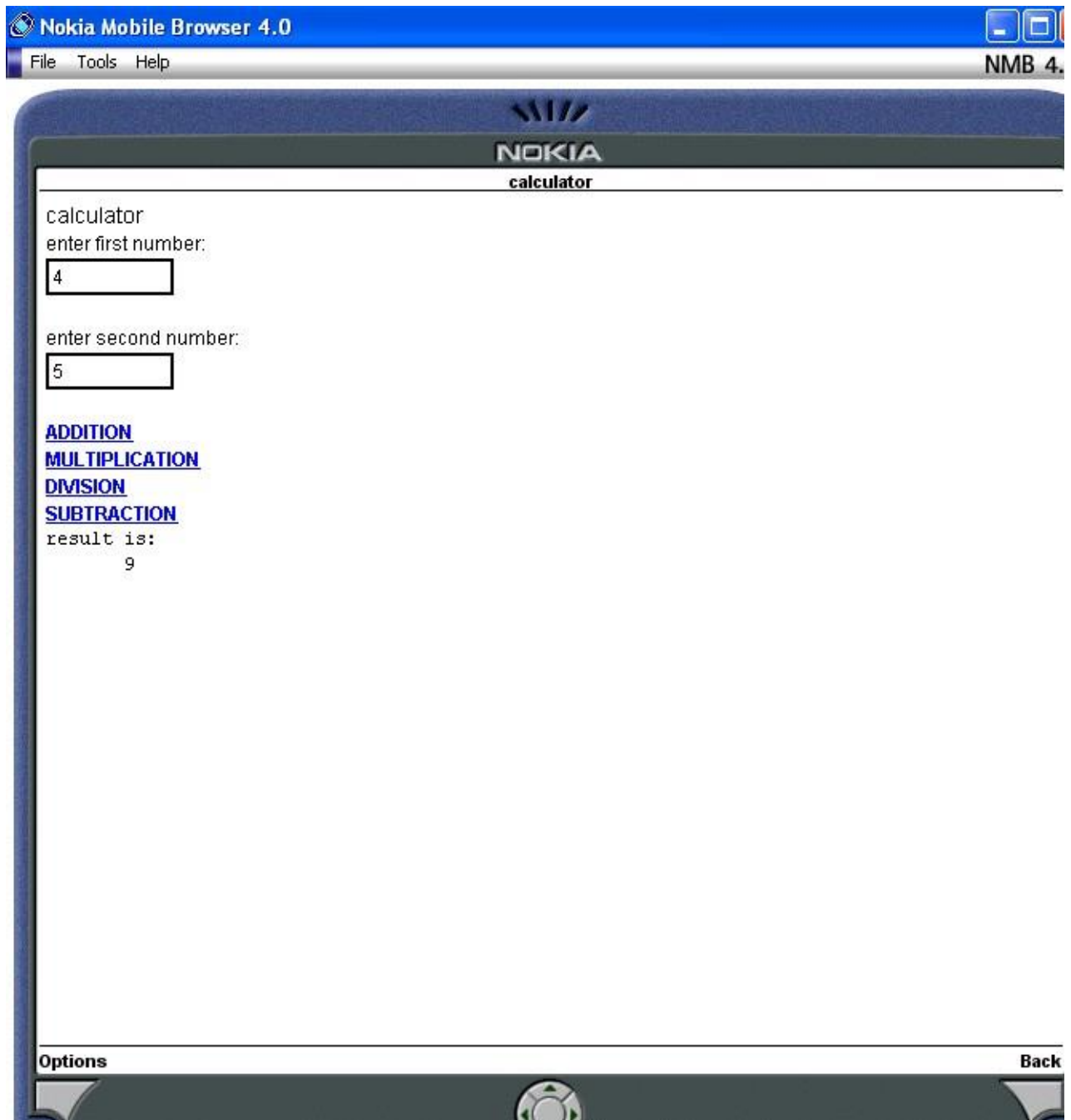
```
second = Lang.parseInt(second);
```

```
WMLBrowser.setVar("result", first / second);
```

```
WMLBrowser.refresh();
```

```
}
```

OUTPUT :



The screenshot displays the Nokia Mobile Browser 4.0 interface. The title bar at the top reads "Nokia Mobile Browser 4.0" and includes a menu bar with "File", "Tools", and "Help". The address bar shows "NMB 4.". The main content area features the Nokia logo and the title "calculator". The application prompts the user to "enter first number:" and "enter second number:". The first input field contains the number "4" and the second contains "5". Below the input fields, there are four menu options: "ADDITION", "MULTIPLICATION", "DIVISION", and "SUBTRACTION". The "result is:" section displays the number "9". At the bottom of the screen, there are "Options" and "Back" buttons.

Nokia Mobile Browser 4.0

File Tools Help

NMB 4.

NOKIA

calculator

calculator

enter first number:

4

enter second number:

5

[ADDITION](#)

[MULTIPLICATION](#)

[DIVISION](#)

[SUBTRACTION](#)

result is:

9

Options Back

