

SETHU INSTITUTE OF TECHNOLOGY
(An Autonomous Institution | Accredited with 'A' Grade by NAAC)
PULLOOR, KARIAPATTI - 626 115



DEPARTMENT OF COMPUTER SCIENCE & DESIGN

2023 - 24

(ODD SEMESTER)

INTERACTIVE DESIGN LABORATORY

21UCD509

NAME :

REG NO :

Head of the Department

End of Semester Practical Examination held on

Internal Examiner

External Examiner

INDEX

Ex. No	Date	Experiment	Marks		Page No.	Signature
			Observation (10)	Viva (10)		
1.		Static Webpages design for an online book store website				
2.		Design a webpage using HTML and CSS for Yoga Schedule				
3.		Design a webpage using HTML and CSS for Online Ticket Booking for ICC World Cup				
4.		To Create a Quiz Applications Program Using HTML, CSS and Javascript				
5.		Write a Javascript to validate the fields of the Login page and Registration page				
6.		Write a HTML Page including required Javascript that takes a number from textfield and displays it in Words				
7.		To Develop a Simple Android Application for a Simple Calculator App				
8.		To Develop a Simple Android Application to display different Shapes				
9.		To Develop a Android Application that implements Multithreading				
10.		To Develop a Simple Android Application to Create and Display the Notification				
11.		To Develop a Simple Android Application to Create an Alarm				
12.		To Develop a Simple Android Application to design a Simple BMI Calculator				

Exercise 1

Online Bookstore:

You are tasked with creating a web page for an online bookstore. The web page should display a list of books along with their titles, authors and prices. Each book should be clickable, and when clicked, it should open a new page that displays detailed information about the book, including a description and its cover image.

Aim:

To create a webpage for an online bookstore to display a list of books along with their titles, authors and prices.

Coding:

home.html:

```
<frameset rows="40%,*">
<frame src="top.html" noresize scrolling="NO" name="topframe">
<frameset cols="15%,*">
<frame src="left.html" noresize scrolling="NO" name="leftframe">
<frame src="right.html" noresize name="rightframe" scrolling="auto">
</frameset>
</frameset>
```

top.html:

```
<html>
<head>
<title>Top Frame</title>
</head>
<body bgcolor="YellowGreen ">


<center>
<marquee bgcolor="yellow" width="650" behavior="alternate">
<font face="Brush Script MT" size="8" color="green"><b><i>Online Book Store</i></b>
</font>
</marquee><br>
<font face="Brush Script" size="6" color="white"><b>Created & Maintained By
```

MRCET

</center>

<table width="100%" height="50%" cellspacing=10>

<tr align="center">

<td>HOME </td>

<td>LOGIN</td>

<td>REGISTER </td>

<td>CATALOGUE</td>

</tr>

</table>

</body>

</html>

left.html:

<html>

<body align="center" bgcolor="bisque">

CSE

ECE

EEE

MECH

</body>

</html>

right.html:

<html>

<body bgcolor="orange">

<center>

<h1>Welcome to the Online Book Store!!!


```

<h2><b> "A Huge Collection Of Engineering E-Books"</b></h2></font>
</center>
</body>
</html>
cse.html:
<html>
<head><title>CSE</title></head>
<body bgcolor="cyan">
<center><font color="blue"><h1>Computer Science and Engineering</h1></font></center>
<br>
<table align="center">
<tr>
<td>Text Books</td>
<td>
<select >
<option value="select the book" selected>Select the book
<option value="C&Ds">C&Ds
<option value="Ads">Ads
<option value="Java">Java
<option value="Oracle">Oracle
<option value="Ms SQL Server">Ms SQL Server
<option value="MySql">MySql
</select>
</td></tr>
<tr>
<td>Quantity</td>
<td><input type="text" id="q"></td>
</tr>
<tr>
<td></td>
<td>
<form method=post action="order.html">
<input type="submit" value=ok />
</form>

```

```

</td>
</tr>
</table>
<center>
<pre> Cost of one book is"500" + shipping "100" </pre>
</center>
</body>
</html>

```

ece.html:

```

<html>
<body bgcolor="Plum">
<h1><font color="blue">Electronics and Communication Engineering</font></h1>
<h2>
<ul>
<li>Digital Circuits</li><li>Signals and Systems</li><li>Digital Communication</li>
</ul>
</h2>
</body>
</html>

```

eee.html:

```

<html>
<body bgcolor="Plum">
<h1><font color="blue">Electrical and Electronics Engineering</font></h1>
<h2>
<ul type="square">
<li>Concepts in Electric Circuits</li>
<li>Introduction to Electronic Engineering</li>
<li>Electrical Power</li>
</ul>
</h2>
</body>
</html>

```

mech.html:

```

<html>

```

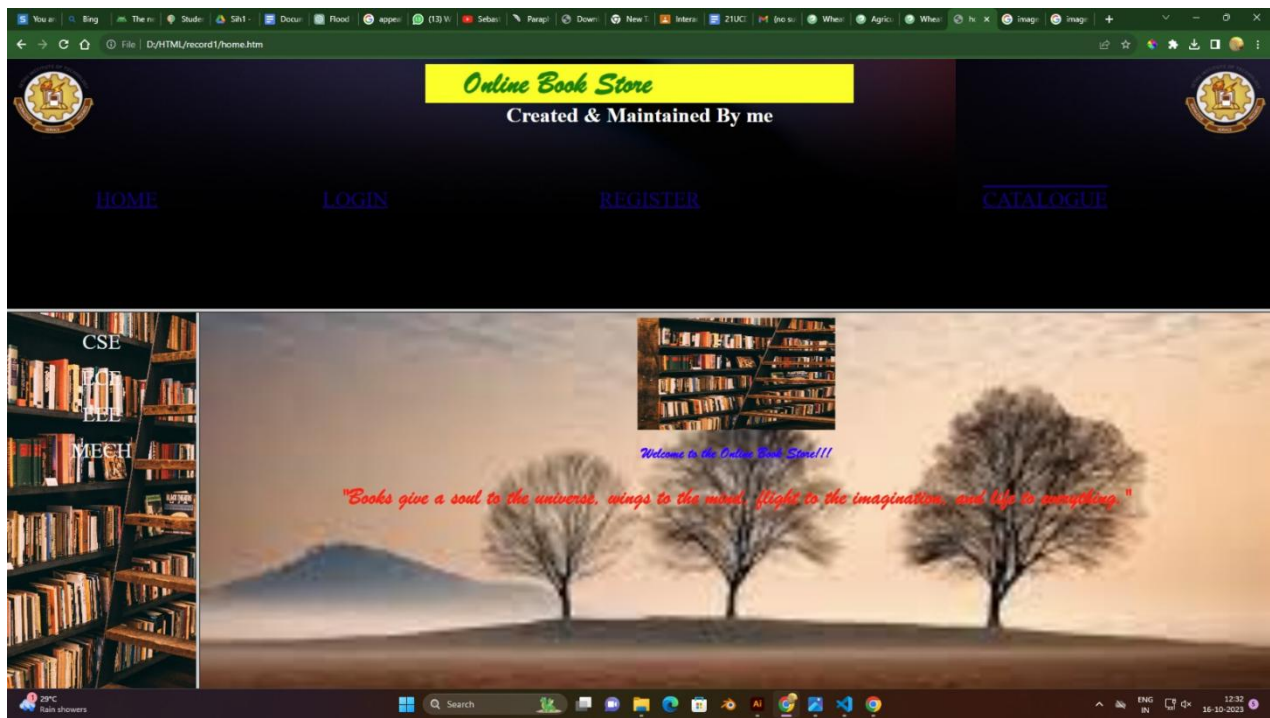

[illegible]**order.html:**

```
<html>
<head><title>order conformation</title></head>
<body bgcolor="cyan">
<center>
<pre><strong>
<b>Your order Is Conformed
</strong></pre>
<h2><b>THANK YOU...Visit Again</h2>
</center>
```


</body>

</html>

Output:



Result:

Thus the webpage for an online bookstore, displaying a list of books along with their titles, authors and prices has been created.

Exercise 2

Design a webpage using HTML and CSS for Yoga Schedule which includes yoga schedule for week days with various Yoga asana names and information about the yoga.

Aim:

To design a webpage using HTML and CSS that incorporates yoga schedule for week days with various yoga asana names and information about the yoga.

Coding:

Backpain.html

```
<html>
<body>
<center>
<h2><font color="purple" face="Calisto MT"><u>YOGA FOR BACK
PAIN</u></font></h2></center><br>
<center>
<p>Does yoga help with back pain? It certainly can, if done correctly. </p>
<p>Yoga can not only help you tune in to your body, but many poses are effective in stretching
and toning the muscles that support your spine.<br>
The result? Less pain, more balance, and better posture and alignment.</p></center>
<h3>1) Bow Pose</h3>
<center>
<p>Bend back into the shape of a bow to feel energetically locked, loaded,<br> and ready to take
aim.</p></center>
<h3>2) Downward-Facing Dog</h3>
<center>
<p>One of yoga's most widely recognized poses, Downward-Facing Dog Pose, called Adho
Mukha Svanasana in Sanskrit,<br>
works to strengthen the core and improve circulation.
This rejuvenating pose works to provide a delicious,<br> full-body stretch.</p>
</center>
<h3>3) Revolved Side Angle Pose</h3>
<center><br>
```

<p>This revolved variation of Utthita Parsvakonasana requires a lot of flexibility
to twist so deeply and ground the back heel.</p>

</center>

<h3>4) Revolved Triangle Pose</h3>

<center>

<p>A counterpose to Utthita Trikonasana and preparation for seated forward bends
 and twists, this pose is key to a skilled practice. </p>

</center>

</body>

</html>

Feedback.html

<html>

<body>

<form action="#" method="POST">

<h3>First Name: <input type="text" placeholder="Your name"></h3>

<h3>Last Name: <input type="text" placeholder="Your last name"></h3>

<h3>E-Mail: <input type="text" placeholder="E mail"></h3>

<h3>Country: <select><option>---</option>

<option>India</option>

<option>Canada</option>

<option>USA</option>

<option>Australia</option>

<option>UK</option>

<option>New Zeland</option>

</select></h3>

<h3>Comments:
<textarea rows="3"cols="60" placeholder="Add your comments"></textarea></h3>

<h3>Ratings :

<input type="radio">1

<input type="radio">2

<input type="radio">3

<input type="radio">4

```

<input type="radio">5</h3>
<br>
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="reset" value="Reset">
</form>
</body>
</html>

```

Fitness.html

```

<html>
<body>
<center>
<h2><font color="purple" face="Calisto MT"><u>YOGA FOR
FITNESS</u></font></h2></center>
<center><br>
<p><h4>Yoga for fitness poses will work your entire body, burn fat, and tone muscle.<br>
Include these poses into your daily routine for overall improvements in strength and
fitness.</h4></p></center><br>
<h3>1) Boat Pose(Paripurna Navasana)</h3>
<center>
<p><h4>An ab and deep hip flexor strengthener, Boat Pose, or Paripurna Navasana in Sanskrit,
requires you to balance on <br>
tripod of your sitting bones and tailbone to build mental and physical focus, inspiring a full-body
awareness.</p></center>
<h3>2) Dolphin Plank Pose</h3>
<center>
<p><h4>A modification of Plank Pose, Dolphin Plank Pose strengthens and tones the<br> core,
thighs, and arms..</p>
</center>
<h3>3) Dolphin Pose</h3>
<center>
<p><h4>Dolphin pose strengthens the core, arms, and legs, while also nicely <br>opening the
shoulders.</p>
</center>
<h3>4) Extended Side Angle Pose</h3>

```

<center>

<p><h4>Find length in your side body, from your heel to your fingertips with
 Extended Side Angle Pose.</p>

</center>

</body>

</html>

Flex.html

<html>

<body>

<center>

<h2><u>YOGA FOR FLEXIBILITY</u></h2></center>

<center>

<p>These yoga poses for flexibility will help you lengthen and stretch your muscles in a safe, effective way. </p>

<p>Include these poses in your practice regularly to see improvements.</p></center>

<h3>1) Half Frog Pose</h3>

<center>

<p>Ease up into Half Frog Pose, called Ardha Bhekasana in Sanskrit.

This pose strengthens the back while gently opening up the shoulders, chest, and thighs—a loving treat for the entire body.</p></center>

<h3>2) Head-to-Knee Forward Bend</h3>

<center>

<p>Janu Sirsasana or Head-to-Knee Forward Bend is appropriate for all levels
of student and a spinal twist to boot.</p>

</center>

<h3>3) Lord of the Dance Pose</h3>

<center>

<p>Want to, like, connect with cosmic energy? Nataraja is another name for Shiva and his dance symbolizes cosmic energy.

Natarajasana, or Lord of the Dance Pose.</p>

</center>

<h3>4) One-Legged King Pigeon Pose</h3>

<center>

<p>One-Legged King Pigeon Pose is a deep backbend that puffs the chest,
 making a yogi resemble a pigeon.</p>

</center>

</body>

</html>

Hbp.html

<html>

<body>

<center>

<h2><u>YOGA FOR HIGH BLOOD
PRESSURE</u></h2></center>

<center>

<p>Improve your circulation by adding these yoga for high blood pressure poses into your daily practice.</h3></p></center>

<h3>1) Bound Angle Pose</h3>

<center>

<p><h3>Bound Angle Pose/Cobbler's Pose, calleed Baddha Konasana in Sanskrit, works to open the deepest part of the hip muscles.

It will gently stretch sore hips post-workout while improving postural and body awareness.</h3></p></center>

<h3>2) Bridge Pose</h3>

<center>

<p><h3>Setu Bandha Sarvangasana can be whatever you need—energizing,
 rejuvenating, or luxuriously restorative.</h3></p>

</center>

<h3>3) Standing Forward Bend</h3>

<center>

<p><h3>Uttanasana will wake up your hamstrings and soothe your mind.</h3></p>

</center>

<h3>4) Reclining Hero Pose</h3>

<center>

<p><h3>Virasana is a balm for tired legs at the end of the day, as well as an alternative
 to Lotus for seated meditation.</h3></p>

</center>

```
</body>
```

```
</html>
```

Stress.html

```
<html>
```

```
<body>
```

```
<center>
```

```
<h2><font color="purple" face="Calisto MT"><u>YOGA FOR  
STRESS</u></font></h2></center><br>
```

```
<center>
```

```
<p>Don't miss Yoga Journal's six-week Yoga for Stress and Anxiety course that'll make a  
<br>lasting change in the way you work, love, and live.</p></center>
```

```
<h3>1) Channel-Cleaning Breath</h3>
```

```
<center>
```

```
<p>Sometimes considered a preparation for pranayama,<br> other times a formal practice in  
itself</p></center>
```

```
<h3>2) Big Toe Pose</h3>
```

```
<center><br>
```

```
<p>This pose gently lengthens and strengthens even stubbornly tight hamstrings.</p>  
</center>
```

```
<h3>3) Child's Pose</h3>
```

```
<center>
```

```
<p>Take a break. Balasana is a restful pose that can be sequenced between <br>more challenging  
asanas.</p>
```

```
</center>
```

```
<h3>4) Plow Pose</h3>
```

```
<center><br>
```

```
<p>Plow Pose reduces backache and can help you get to sleep.</p>  
</center>
```

```
</body>
```

```
</html>
```

Yoga.html

```
<html>
```

```
<body><style>
```

```

td,th,table{
text-align:left-center-left;
font-family:arial;
border-radius:8px;
}
button{background:gray;
font-family:arial;
border-radius:8px;
}
</style>
<center>

<h3><font color="purple" face="Calisto MT"><u>YOGA ASANAS WORKOUT PLANS
</u></font></h3>
<table border="2" cellpadding="3" cellspacing="3">
<tr bgcolor="#12AD2B">
<th>Days</th>
<th>Yoga</th>
<th>Yoga Asanas Name</th>
<th>Informations</th>
</tr>
<tr bgcolor="gold">
<td>Monday</td>
<td>YOGA FOR FITNESS</td>
<td>Boat Pose,Dolphin Plank Pose,Dolphin Pose,Extended Side Angle Pose</td>
<td><a href="fitness.html">View More</a></td>
</tr>
<tr bgcolor="gold">
<td>Tuesday</td>
<td>YOGA FOR HIGH BLOOD PRESSURE</td>
<td>Bound Angle Pose,Bridge Pose,Standing Forward Bend,Reclining Hero Pose</td>
<td><a href="hbp.html">View More</a></td>
</tr>
<tr bgcolor="gold">

```



```

<td>Wednesday</td>
<td>YOGA FOR BACK PAIN</td>
<td>Bow Pose,Downward-Facing Dog,Revolved Side Angle Pose,Revolved Triangle Pose</td>
<td><a href="backpain.html">View More</a></td>
</tr>
<tr bgcolor="gold">
<td>Thursday</td>
<td>YOGA FOR STRESS</td>
<td>Channel-Cleaning Breath,Big Toe Pose,Child's Pose,Plow Pose</td>
<td><a href="stress.html">View More</a></td>
</tr>
<tr bgcolor="gold">
<td>Friday</td>
<td>YOGA FOR FLEXIBILITY</td>
<td>Half Frog Pose,Head-to-Knee Forward Bend,Lord of the Dance Pose,<br>One-Legged King
Pigeon Pose</td>
<td><a href="flex.html">View More</a></td>
</tr>
</table><br><br><br>
<Button><a href="feedback.html">Feedback</a></button>
</center>
</body>
</html>

```

Output:

Backpain.html x Feedback.html x +

File | D:\Gogo\experiment%202\Feedback.html

First Name:

Last Name:

E-Mail:

Country:

Comments:

Ratings : ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5



YOGA ASANAS WORKOUT PLANS

Days	Yoga	Yoga Asanas Name	Informations
Monday	YOGA FOR FITNESS	Boat Pose,Dolphin Plank Pose,Dolphin Pose,Extended Side Angle Pose	View More
Tuesday	YOGA FOR HIGH BLOOD PRESSURE	Bound Angle Pose,Bridge Pose,Standing Forward Bend,Reclining Hero Pose	View More
Wednesday	YOGA FOR BACK PAIN	Bow Pose,Downward-Facing Dog,Revolved Side Angle Pose,Revolved Triangle Pose	View More
Thursday	YOGA FOR STRESS	Channel-Cleaning Breath,Big Toe Pose,Child's Pose,Plow Pose	View More
Friday	YOGA FOR FLEXIBILITY	Half Frog Pose,Head-to-Knee Forward Bend,Lord of the Dance Pose,One-Legged King Pigeon Pose	View More

[Feedback](#)


Backpain.html x Feedback.html x Fitness.html x Flex.html x Hip.html x Stress.html x Yoga.html

File | D:\Gojo\experiment%202\Backpain.html

YOGA FOR BACK PAIN


Does yoga help with back pain? It certainly can, if done correctly.
Yoga can not only help you tune in to your body, but many poses are effective in stretching and toning the muscles that support your spine.
The result? Less pain, more balance, and better posture and alignment.

1) Bow Pose



Bend back into the shape of a bow to feel energetically locked, loaded, and ready to take aim.

2) Downward-Facing Dog



One of yoga's most widely recognized poses, Downward-Facing Dog Pose, called Adho Mukha Svanasana in Sanskrit, works to strengthen the core and improve circulation. This rejuvenating pose works to provide a delicious, full-body stretch.

3) Revolved Side Angle Pose

Tomorrow's low
Near record

Search

ENG IN 12:07 16-10-2023


Backpain.html x Feedback.html x Fitness.html x Flex.html x Hip.html x Stress.html x Yoga.html

File | D:\Gojo\experiment%202\Fitness.html

YOGA FOR FITNESS


Yoga for fitness poses will work your entire body, burn fat, and tone muscle.
Include these poses into your daily routine for overall improvements in strength and fitness.

1) Boat Pose(Paripurna Navasana)



An ab and deep hip flexor strengthener, Boat Pose, or Paripurna Navasana in Sanskrit, requires you to balance on tripod of your sitting bones and tailbone to build mental and physical focus, inspiring a full-body awareness.

2) Dolphin Plank Pose



A modification of Plank Pose, Dolphin Plank Pose strengthens and tones the core, thighs, and arms.

3) Dolphin Pose

29°C
Rain showers

Search

ENG IN 12:08 16-10-2023


Backpain.html x Feedback.html x Fitness.html x Flex.html x Hbp.html x Stress.html x Yoga.html x +

File | D:\Gogo\experiment%202\Flex.html

YOGA FOR FLEXIBILITY


These yoga poses for flexibility will help you lengthen and stretch your muscles in a safe, effective way.
Include these poses in your practice regularly to see improvements.

1) Half Frog Pose




Ease up into Half Frog Pose, called Ardha Bhakasana in Sanskrit.
This pose strengthens the back while gently opening up the shoulders, chest, and thighs—a loving treat for the entire body.

2) Head-to-Knee Forward Bend



Janu Sirasana or Head-to-Knee Forward Bend is appropriate for all levels of student and a spinal twist to boot.

3) Lord of the Dance Pose



29°C Rain showers

Search

ENG IN 12:12 16-10-2023


Backpain.html x Feedback.html x Fitness.html x Flex.html x Hbp.html x Stress.html x Yoga.html x +

File | D:\Gogo\experiment%202\Hbp.html

YOGA FOR HIGH BLOOD PRESSURE


Improve your circulation by adding these yoga for high blood pressure poses into your daily practice.

1) Bound Angle Pose



Bound Angle Pose/Cobbler's Pose, called Baddha Konasana in Sanskrit, works to open the deepest part of the hip muscles.
It will gently stretch sore hips post-workout while improving postural and body awareness.

2) Bridge Pose



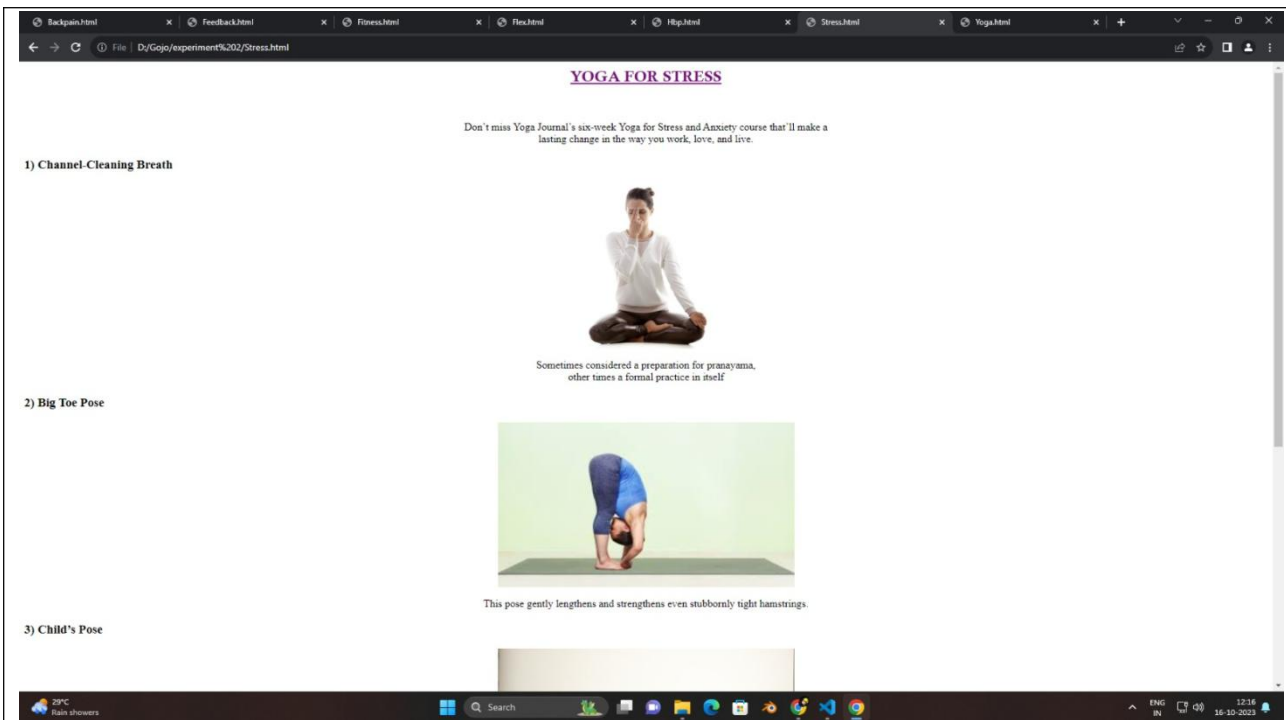
Setu Bandha Sarvangasana can be whatever you need—energizing, rejuvenating, or luxuriously restorative.

3) Standing Forward Bend

29°C Rain showers

Search

ENG IN 12:13 16-10-2023



Result:

Thus a webpage using HTML and CSS that incorporates yoga schedule for week days with various yoga asana names and information about the yoga has been designed.

Exercise 3

Design a website which includes Ticket registration, match Schedule and match venue for ICC World cup Cricket using HTML and CSS

Aim:

To design a website which displays match Schedule, match venue and ticket registration for ICC World cup Cricket using HTML and CSS

Coding:

Style.css

```
*
{
padding:0;
margin:0;
}
ul
{
list-style:none;
background: blue;
}
ul li
{
display:inline-block;
position:relative;
}
ul li a
{
display:block;
padding :20px; 25px;
color:white;
text-decoration: none;
text-align:center;
font-size:20px;
}
ul li ul.dropdown li
```

```
{
display:block;
}
ul li ul.dropdown
{
width:100%;
background:blue;
position:absolute;
z-index:999;
display:none;
}
ul li a:hover
{
background:black;
}
ul li:hover ul.dropdown
{
display:block;
}
ul li ul.team li
{
display:block;
}
ul li ul.team
{
width:100%;
background:blue;
position:absolute;
z-index:999;
display:none;
}
ul li a:hover
{
background:black;
```

```

}
ul li:hover ul.team
{
display:block;
}
Worldcup.html
<html>
<head>
<title>ICC T20 World Cup 2021</title>
<link rel="stylesheet" href="style/style.css"></head>
<body>

<center>
<br>
<br>
<nav>
<ul>
<li><a href="#">Home</a></li>
<li><a href="ticket.html">Ticket Registration</a></li>
<li><a href="venue.html">Match venue</a>
<ul class="dropdown">
<li><a href="venue.html">Dubai</a></li>
<li><a href="venue.html">Abu Dhabi</a></li>
<li><a href="venue.html">Sharjah</a></li>
</ul>
</li>
<li><a href="#">Teams Squard</a>
<ul class="team">
<li><a href="indiasqu.html">India</a></li>
<li><a href="#">New Zealand</a></li>
<li><a href="#">Pakistan</a></li>
<li><a href="#">Afghanistan</a></li>
</ul>
</li>

```



```

</ul>
</nav>
<br>
<h3><font color="purple" face="Calisto MT"><u>ICC T20 World Cup 2021</u></font></h3>
<table border="2">
<tr>
<th>Tournament</th>
<th>T20 World Cup</th>
</tr>
<tr>
<td>Year</td>
<td>2021</td>
</tr>
<tr>
<td>Match Type</td>
<td>Twenty-20</td>
</tr>
<tr>
<td>Host Country</td>
<td>UAE</td>
</tr>
<tr>
<td>Organizer</td>
<td>ICC</td>
</tr>
<tr>
<td>Starting Date</td>
<td>17 Oct 2021</td>
</tr>
<tr>
<td>Ending Date</td>
<td>14 Nov 2021</td>
</tr>
<tr>

```

```

<td>Match Schedule</td>
<td>Check Below</td>
</tr>
</table>
<br>
<h3><font color="purple" face="Calisto MT"><u>T20 World Cup Schedule
2021</u></font></h3>
<table border="2">
<tr>
<th>Mathes</th>
<th>Venue</th>
<th>Date & Time</th>
</tr>
<tr>
<td>India Vs Pakistan</td>
<td>Dubai</td>
<td>24 Oct , 7:30 Pm</td>
</tr>
<tr>
<td>Afghanistan Vs B1</td>
<td>Sharjah</td>
<td>25 Oct , 7:30 Pm</td>
</tr>
<tr>
<td>Pakistan Vs New Zeland</td>
<td>Sharjah</td>
<td>26 Oct , 7:30 Pm</td>
</tr>
<tr>
<td>B1 Vs A2</td>
<td>Abu Dhabi</td>
<td>27 Oct , 7:30 Pm</td>
</tr>
<tr>

```

<td>Afghanistan Vs Pakistan</td>

<td>Dubai</td>

<td>29 Oct , 7:30 Pm</td>

</tr>

<tr>

<td>Afghanistan Vs A2</td>

<td>Abu Dhabi</td>

<td>31 Oct , 3:30 Pm</td>

</tr>

<tr>

<td>India Vs New Zeland</td>

<td>Dubai</td>

<td>31 Oct , 7:30 Pm</td>

</tr>

<tr>

<td>A2 Vs Pakistan</td>

<td>Abu Dhabi</td>

<td>2 Nov , 7:30 Pm</td>

</tr><tr>

<td>B1 Vs New Zeland</td>

<td>Dubai</td>

<td>3 Nov , 3:30 Pm</td>

</tr><tr>

<td>India Vs Afghanistan</td>

<td>Abu Dhabi</td>

<td>3 Nov , 7:30 Pm</td>

</tr><tr>

<td>New Zeland Vs A2</td>

<td>Sharjah</td>

<td>5 Nov , 3:30 Pm</td>

</tr><tr>

<td>India Vs B1</td>

<td>Dubai</td>

<td>5 Nov , 7:30 Pm</td>

```

</tr><tr>
<td>New Zeland Vs Afghanistan</td>
<td>Abu Dhabi</td>
<td>7 Nov , 3:30 Pm</td>
</tr><tr>
<td>Pakistan Vs B1</td>
<td>Sharjah</td>
<td>7 Nov , 7:30 Pm</td>
</tr><tr>
<td>India Vs A2</td>
<td>Dubai</td>
<td>8 Nov , 7:30 Pm</td>
</tr>
</table><br>
<h3><font color="purple" face="Calisto MT"><u>Qualifier Rounds</u></font></h3>
<table border="2">
<tr>
<th>Date</th>
<th>Team</th>
<th>Match Name</th>
<th>Venue</th>
</tr>
<tr>
<td>10 Nov 2021</td>
<td>TBC Vs TBC</td>
<td>1st Semi-Finals</td>
<td>Abu Dhabi</td>
</tr>
<tr>
<td>11 Nov 2021</td>
<td>TBC Vs TBC</td>
<td>2st Semi-Finals</td>
<td>Dubai</td>
</tr>

```

```

<tr>
<td>14 Nov 2021</td>
<td>TBC Vs TBC</td>
<td>Finals</td>
<td>Dubai</td>
</tr>
</table>
</body>
</html>

```

Venue.html

```

<html>
<body>
<head><title>Venue</title>
</head>
<body><center>
</center>
  <map name="imagemap">
<area shape="rect" coords="415,330,468,360"
href="https://goo.gl/maps/aZ9Qy12RKBrYNvnY8" >
<area shape="rect" coords="508,228,572,258"
href="https://goo.gl/maps/yF9MV3a9zaXW9VWA7" >
<area shape="rect" coords="539,185,585,215" href="https://g.page/SCCSCS?share" >
  </map>
</body>
</html>

```

Ticket.html

```

<html>
<body bgcolor="gold">
<br>
<h1 align="center"><font face="Times New Roman" color="green"><u>TICKET
REGISTRATION</u></font></h1><br>
<h3>Name : <input type="text" placeholder="Enter First name" size="20"></h3>
<h3>Mobile No : <input type="text" placeholder="Enter Mobile No" size="16"></h3>
<h3>Gender : <input type="radio" >Male

```

```

<input type="radio" name="male" value="Male">Female
<table border="2" align="center">
<tr>
<th>No Of Tickets</th>
<th>Normal Price for 1 Person</th>
<tr>
<tr>
<td><select><option>1
<option>2
<option>3
<option>4
<option>5
<option>6
<option>7
</select>
<td>1300</td>
</tr>
</table>
<br>
<h1><font face="Times New Roman" color="green"><u>Payment Option</u></font></h1><br>
<input type="radio">Credit/Debit Card
<input type="radio">UPI
<input type="radio">Net Banking
<br>
<center><input type="submit" name="Submit" value="Submit">
<input type="reset" name="reset" value="Reset"></center>
</body>
</html>

```

Indiasqu.html

```

<html>
<body bgcolor="navy"><center>
<h3><font color="#ffa500" face="Calisto MT"><u>INDIA T20 World Cup 2021
SQUARD</u></font></h3>
</center>

```

</body>

</html>

Output:

ICC T20 World Cup 2021

File | D:/html/Worldcup.html#

ICC CRICKET WORLD CUP

Home Ticket Registration Match venue Teams Squad

ICC T20 World Cup 2021

Tournament	T20 World Cup
Year	2021
Match Type	Twenty-20
Host Country	UAE
Organizer	ICC
Starting Date	17 Oct 2021
Ending Date	14 Nov 2021
Match Schedule	Check Below

T20 World Cup Schedule 2021

Matches	Venue	Date & Time
India Vs Pakistan	Dubai	24 Oct , 7:30 Pm
Afghanistan Vs B1	Sharjah	25 Oct , 7:30 Pm
Pakistan Vs New Zealand	Sharjah	26 Oct , 7:30 Pm
B1 Vs A2	Abu Dhabi	27 Oct , 7:30 Pm
Afghanistan Vs Pakistan	Dubai	29 Oct , 7:30 Pm
Afghanistan Vs A2	Abu Dhabi	31 Oct , 3:30 Pm
India Vs New Zealand	Dubai	31 Oct , 7:30 Pm
A2 Vs Pakistan	Abu Dhabi	2 Nov , 7:30 Pm
B1 Vs New Zealand	Dubai	3 Nov , 3:30 Pm
India Vs Afghanistan	Abu Dhabi	3 Nov , 7:30 Pm
New Zealand Vs A2	Sharjah	5 Nov , 3:30 Pm
India Vs B1	Dubai	5 Nov , 7:30 Pm
New Zealand Vs Afghanistan	Abu Dhabi	7 Nov , 3:30 Pm
Pakistan Vs B1	Sharjah	7 Nov , 7:30 Pm
India Vs A2	Dubai	8 Nov , 7:30 Pm

Qualifier Rounds

Date	Team	Match Name	Venue
10 Nov 2021	TBC Vs TBC	1st Semi-Finals	Abu Dhabi
11 Nov 2021	TBC Vs TBC	2nd Semi-Finals	Dubai
14 Nov 2021	TBC Vs TBC	Finals	Dubai

Ticket.html

File | D:/html/ticket.html

TICKET REGISTRATION

Name :

Mobile No :

Gender : ☒ Male ☐ Female

No Of Tickets


Normal Price for 1 Person

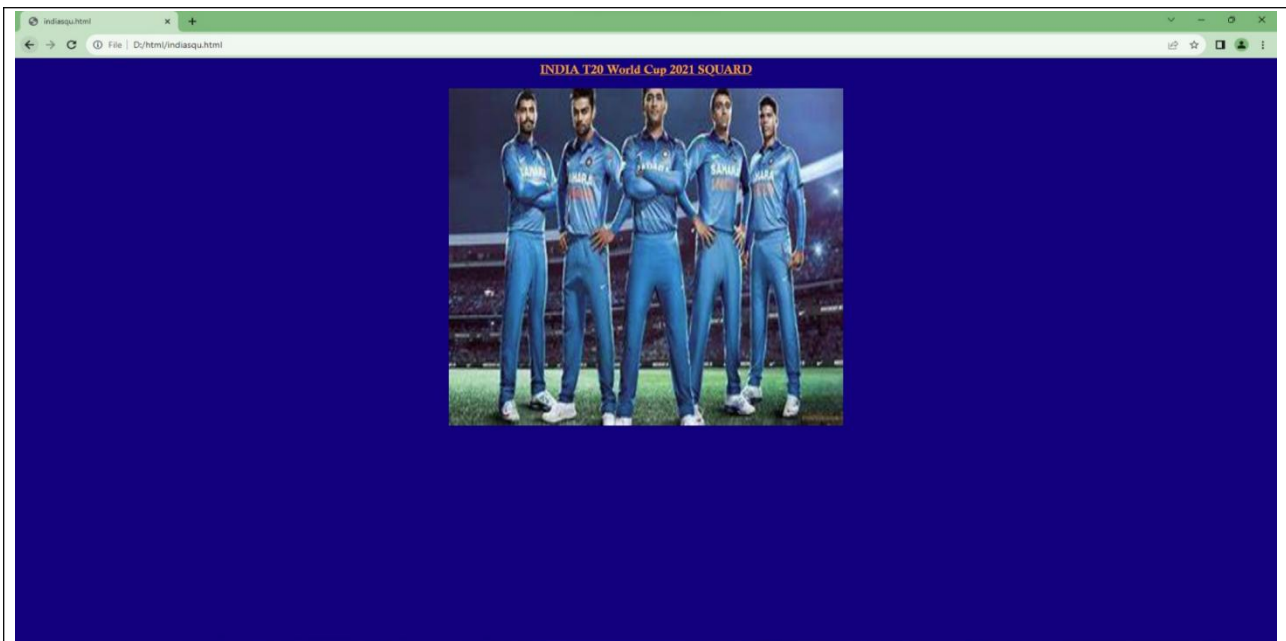
Payment Option

☐ Credit/Debit Card ☐ UPI ☐ Net Banking

Venue

File | D:/html/venue.html



**Result:**

Thus a website which displays match Schedule, match venue and ticket registration for ICC World cup Cricket has been designed using HTML and CSS.

Exercise 4

Quiz Application Program:

Create a web-based quiz application using JavaScript. The application should present a series of questions to the user, allow them to select an answer, and provide immediate feedback on whether the answer was correct or not. At the end of the quiz, the application should display the user's score.

Aim:

To create a web-based quiz application using Java Script.

Coding:

index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="panel">

<h1>Quiz Application Using JavaScript</h1>

<div class="question" id="ques"></div>

<div class="options" id="opt"></div>

<button onclick="checkAns()" id="btn">SUBMIT</button>

<div id="score"></div>
```

```
<script src="script.js"></script>
```

```
</div>
```

```
</body>
```

```
</html>
```

style.css

```
body {
```

```
    background-color: aliceblue;
```

```
}
```

```
.panel {
```

```
    margin-top: 8%;
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    align-items: center;
```

```
    justify-content: center;
```

```
    color: navy;
```

```
}
```

```
.question {
```

```
    font-size: 30px;
```

```
    margin-bottom: 20px;
```

```
}
```

```
.options {
```

```
    font-size: 20px;
```

```
display: grid;

grid-template-columns: repeat(2, 1fr);

grid-gap: 20px;

margin-top: 10px;

margin-bottom: 20px;

position: fixed;

top: 40%;

left: 40%;

}

button {

margin-right: 75px;

margin-top: 8%;

font-size: 20px;

padding: 10px 20px;

background-color: #4f98c2;

color: white;

border: none;

cursor: pointer;

}

#score {

font-size: 30px;

color: darkslategray;
```

```
}
```

Script.js

```
const Questions = [{
```

```
  q: "What is capital of India?",
```

```
  a: [{ text: "Gandhinagar", isCorrect: false },
```

```
    { text: "Surat", isCorrect: false },
```

```
    { text: "Delhi", isCorrect: true },
```

```
    { text: "Mumbai", isCorrect: false }]
```

```
],
```

```
},
```

```
{
```

```
  q: "What is the capital of Thailand?",
```

```
  a: [{ text: "Lampang", isCorrect: false, isSelected: false },
```

```
    { text: "Phuket", isCorrect: false },
```

```
    { text: "Ayutthaya", isCorrect: false },
```

```
    { text: "Bangkok", isCorrect: true }]
```

```
]
```

```
},
```

```
{
```

```
  q: "What is the capital of Gujarat",
```

```
  a: [{ text: "Surat", isCorrect: false },
```

```
    { text: "Vadodara", isCorrect: false },
```

```

    { text: "Gandhinagar", isCorrect: true },

    { text: "Rajkot", isCorrect: false }

  ]
}

]

let currQuestion = 0

let score = 0

function loadQues() {

  const question = document.getElementById("ques")

  const opt = document.getElementById("opt")

  question.textContent = Questions[currQuestion].q;

  opt.innerHTML = ""

  for (let i = 0; i < Questions[currQuestion].a.length; i++) {

    const choicesdiv = document.createElement("div");

    const choice = document.createElement("input");

    const choiceLabel = document.createElement("label");

    choice.type = "radio";

    choice.name = "answer";

    choice.value = i;

    choiceLabel.textContent = Questions[currQuestion].a[i].text;

    choicesdiv.appendChild(choice);

    choicesdiv.appendChild(choiceLabel);
  }
}

```

```

    opt.appendChild(choicesdiv);

}

}

loadQues();

function loadScore() {

    const totalScore = document.getElementById("score")

    totalScore.textContent = `You scored ${score} out of ${Questions.length}`

}

function nextQuestion()

{

    if (currQuestion < Questions.length - 1) {

        currQuestion++;

        loadQues();

    } else {

        document.getElementById("opt").remove()

        document.getElementById("ques").remove()

        document.getElementById("btn").remove()

        loadScore();

    }

}

function checkAns() {

    const selectedAns = parseInt(document.querySelector('input[name="answer"]:checked').value);

```

```

if (Questions[currQuestion].a[selectedAns].isCorrect) {

    score++;

    console.log("Correct")

    nextQuestion();

} else {

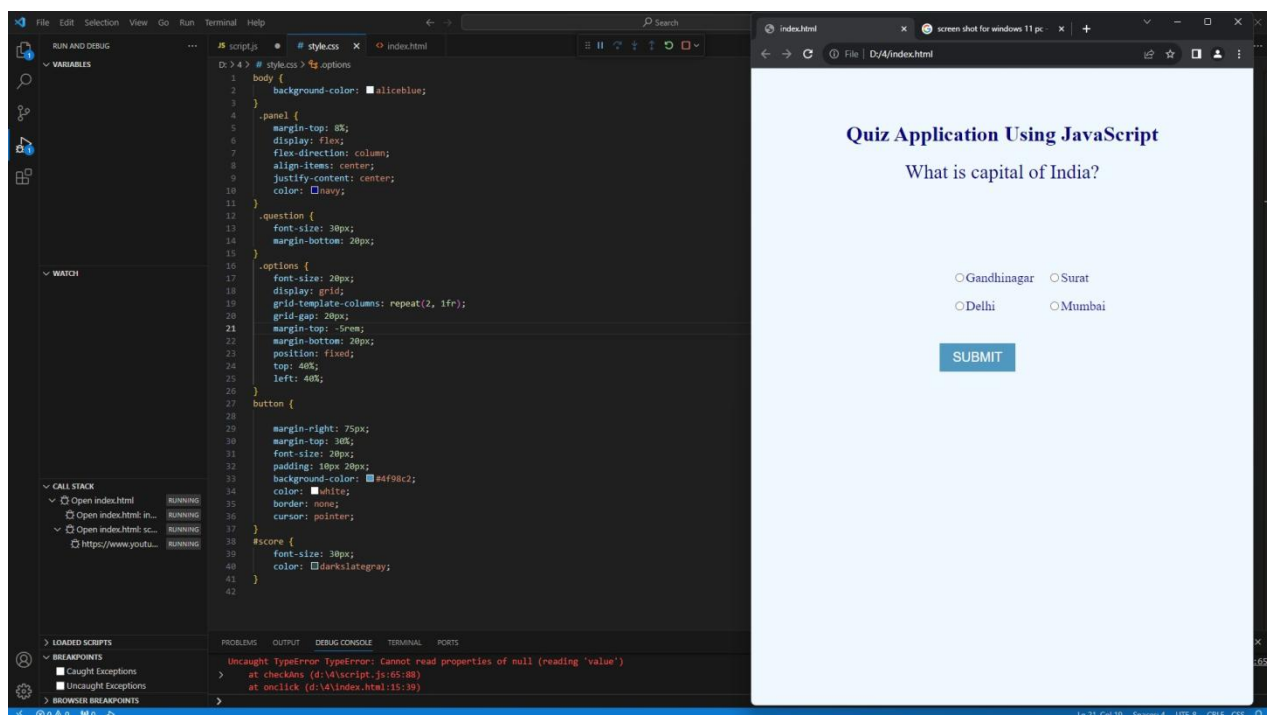
    nextQuestion();

}

}

```

Output:



Result:

Thus a web-based quiz application using Java Script has been developed.

Exercise 5

JavaScript Field Validation for Login and Registration Pages

You want to implement field validation using JavaScript for both the login and registration pages of a website. The validation should ensure that the required fields are not empty, and the user has entered a valid email address.

Aim:

To implement a field validation using Java Script for both login and registration pages of a website.

Coding:

registration.html

```
<html>
<head>
<script type="text/javascript">
function checkfield(){
if(document.regfrm.fname.value.length==0){
alert("fill firstname field")
document.regfrm.fname.focus();
return;
}
if(document.regfrm.lname.value.length==0){
alert("fill lastname field")
document.regfrm.lname.focus();
return;
}
if(document.regfrm.eid.value.length==0){
alert("fill email-id")
document.regfrm.eid.focus();
return;
}
if(document.regfrm.zip.value.length==0){
alert("fill zip field")
document.regfrm.zip.focus();
```



```

return;
}
if(document.regfrm.phonenumber.value.length==0){
alert("fill phone number field")
document.regfrm.phonenumber.focus();
return;
}
else{
alert("Succesfull");
}
}
</script>
</head>
<body>
<center>
<form name="regfrm" method="post">
<p align="center">
<b>
<i><font face="monotype corsiva" color="black" size="7">
<u>Personal Registration Form</u>
</font>
</i>
</b>
</p>
<p align="center">
<font face="arial" color="bluishgreen" >
<b>You must complete fields marked with </b></font>
<font color="bluishgreen" face="times newroman">
*(star)</font></p>
<table border="8" cellpadding="0" cellspacing="0" width="73%" height="10">
<tr><td width="43%" height="22" >Firstname</font></td>
<td width="57%" height="22">
<input type="text" name="fname" size="20">
<font color="red"> * </font></td>

```

```

</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnew roman" size="4">
Lastname
</font>
</td>
<td width="57%" height="22">
<input type="text" name="lname" size="20">
<font color="red"> * </font>
</td>
</tr>
<tr><td width="43%" height="22" >
<font face="timesnewroman" size="4">
Jobtitle
</font>
</td>
<td width="57%" height="22">
<font face="timesnewroman" size="20" >
<select size="1" name="jobsinfo">
<option value>choose..</option>
<option value="Engineer">Engineer</option>
<option value="Lecturer">Lecturer</option>
</select></font>
</td>
</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnewroman" size="4">
Industry
</font>
</td>
<td width="57%" height="22">
<font face="timesnewroman"size="20" >

```

```

<select size="1" name="industry">
<option value>Choose..</option>
<option value="Academic">Academic</option>
<option value="Corporate">Corporate</option>
<option value="Financial">Financial</option>
<option value="Government">Government</option>
<option value="Military">Military</option>
<option value="Research">Research</option>
<option value="Technology">Technology</option>
<option value="Transportation">Transportation</option>
</select></font>
</td>
</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnewroman" size="4">
Email-id
</font>
</td>
<td width="57%" height="22">
<input type="text" name="eid" size="20">
<font color="red"> * </font>
</td>
</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnewroman" size="4">
Country
</font>
</td>
<td width="57%" height="38">
<font face="timesnewroman" size="20">
<select size="1" name="country">
<option value>Choose..</option>

```

```

<option value="IND">INDIA</option>
<option value="US">AMERICA</option>
<option value="JPN">JAPAN</option>
<option value="AUS">AUSTRALIA</option>
<option value="FRN">FRANCE</option>
</select></font>
</td>
</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnewroman" size="4">
Organization
</font>
</td>
<td width="57%" height="22">
<input type="text" name="org" size="20">
</td>
</tr>
<tr>
<td ><font face="timesnewroman" size="4">
Address
</font>
</td>
<td>
<textarea rows="2" cols="16"></textarea>
</td>
</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnewroman" size="4">
Town/city
</font>
</td>
<td width="57%" height="22">

```

```

<input type="text" name="city" size="20">
</td>
</tr>
<tr>
<td width="43%" height="22" >
<font face="timesnewroman" size="4">
State
</font>
</td>
<td width="57%" height="38">
<input type="text" name="state" size="20">
</td>
</tr>
<tr>
<td width="43%" height="38" >
<font face="timesnewroman" size="4">
zip/Postalcode
</font>
</td>
<td width="57%" height="38">
<input type="text" name="zip" size="20">
<font color="red">
*
</font>
</td>
</tr>
<tr>
<td width="43%" height="38" >
<font face="timesnewroman" size="4">
Phonenumber
</font>
</td>
<td width="57%" height="38">
<input type="text" name="phonenumber" size="20">

```

```

<font color="red">
*
</font>
</td>
</tr>
</table>
<p align="center"><font face="timesnewroman" color="bluishgreen">
<input type="radio" name="submit">
I agree terms and Conditions<br>
<input type="button" value="submit" name="b1" onclick="checkfield()">
<input type="reset" value="cancel" name="b2">
</p>
</form>
</center>
</body>
</html>

```

Output:

Personal Registration Form

You must complete fields marked with *(star)

Firstname		*	
Lastname		*	
Jobtitle	choose. ▾		
Industry	Choose. ▾		
Email-id		*	
Country	Choose. ▾		
Organization			
Address			
Town/city			
State			
zip/Postalcode		*	
Phonenumber		*	

☐ I agree terms and Conditions

Result:

Thus a field validation using Java Script for both login and registration pages of a website has been implemented.

Exercise 6

Create a HTML page that includes a text field where the user can enter a number. Upon clicking a button, the entered number should be converted to words and displayed on the page.

Aim:

To create a HTML page that includes a text field where the user can enter a number and upon clicking the button, the entered number should be converted to words and displayed on the page.

Coding:

```
<html>
<head>
<title>HTML - Convert numbers to words using JavaScript</title>
<SCRIPT language=Javascript>
<!--
function isNumberKey(evt)
{
var charCode = (evt.which) ? evt.which : evt.keyCode;
if (charCode != 46 && charCode > 31
&& (charCode < 48 || charCode > 57))
return false;
return true;
}
//-->
</SCRIPT>
<script>
function NumToWord(inputNumber, outputControl)
{
var str = new String(inputNumber)
var splt = str.split("");
var rev = splt.reverse();
var once = ['Zero', ' One', ' Two', ' Three', ' Four', ' Five', ' Six', ' Seven', ' Eight', ' Nine'];
var twos = ['Ten', ' Eleven', ' Twelve', ' Thirteen', ' Fourteen', ' Fifteen', ' Sixteen', '
```

```

Seventeen', ' Eighteen', ' Nineteen'];

var tens = ['', 'Ten', ' Twenty', ' Thirty', ' Forty', ' Fifty', ' Sixty', ' Seventy', ' Eighty', '
Ninety'];
numLength = rev.length;
var word = new Array();
var j = 0;
for (i = 0; i < numLength; i++) {
switch (i) {
case 0:
if ((rev[i] == 0) || (rev[i + 1] == 1)) {
word[j] = "";
}
else {
word[j] = once[rev[i]];
}
word[j] = word[j];
break;
case 1:
aboveTens();
break;
case 2:
if (rev[i] == 0) {
word[j] = "";
}
else if ((rev[i - 1] == 0) || (rev[i - 2] == 0)) {
word[j] = once[rev[i]] + " Hundred ";
}
else {
word[j] = once[rev[i]] + " Hundred and";
}
break;
default: break;
}
j++;

```

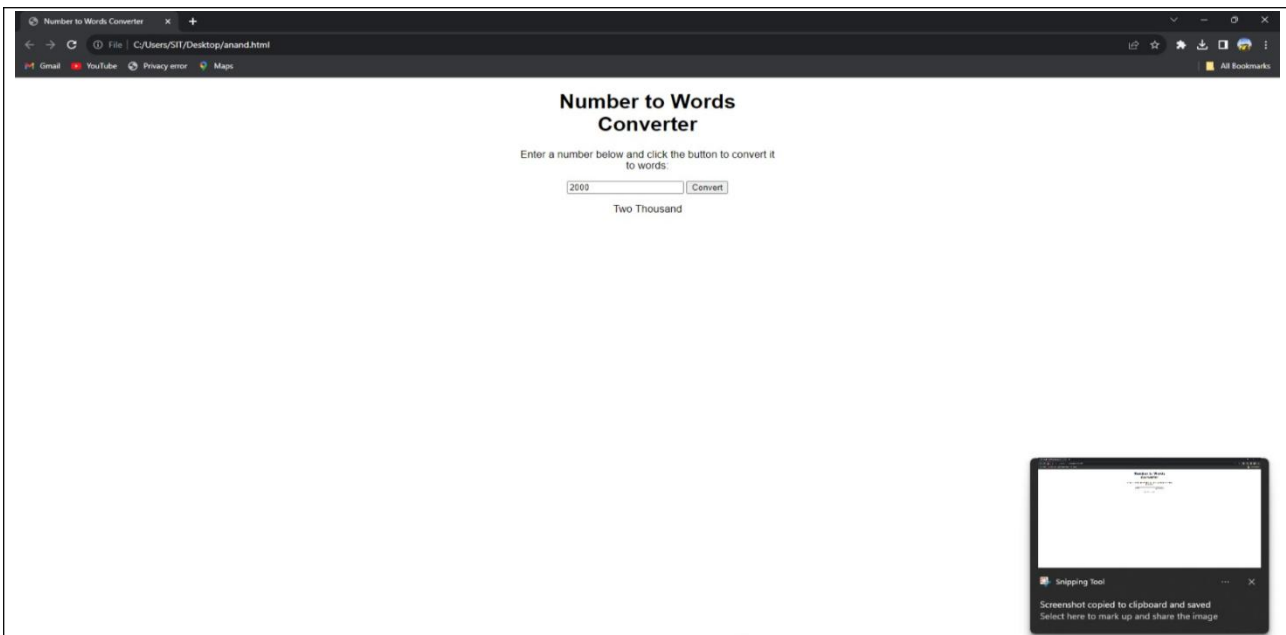


```

}
function aboveTens() {
if (rev[i] == 0) { word[j] = ""; }
else if (rev[i] == 1) { word[j] = twos[rev[i - 1]]; }
else { word[j] = tens[rev[i]]; }
}
word.reverse();
var finalOutput = "";
for (i = 0; i < numLength; i++) {
finalOutput = finalOutput + word[i];
}
document.getElementById(outputControl).innerHTML = finalOutput;
}
</script>
</head>
<body>
<h1>HTML - Convert numbers to words using JavaScript</h1>
<input id="Text1" type="text" onkeypress="return isNumberKey(event)"
onkeyup="NumToWord(this.value,'divDisplayWords');" maxlength="3" style="backgroundcolor:
#efefef; border: 2px solid #CCCCC; font-size: large" />
<br /><br />
<div id="divDisplayWords" style="font-size: 30; color: Teal; font-family: Arial;">
</div>
</body>
</html>

```

Output:



Result:

Thus a HTML page that includes a text field where the user can enter a number has been created. Also, upon clicking the button, the entered number has been converted to words and displayed on the page.

Exercise 7

Implement a simple calculator application for Android. The application should allow users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division.

Aim:

To develop a Simple Android Application to design a Simple Calculator App.

Procedure:

Creating a New project:

- Open Android Studio and then click on **File -> New -> New project**.
- Then type the Application name as “CalcApp”, change the project location and click **Next**.
- Then select the **Minimum SDK** as shown below and click **Next**.
- Then select the **Empty Activity** and click **Next**.
- Finally click **Finish**.

Designing layout for the Android Application:

- Click on **app -> res -> layout -> activity_main.xml**.
- Now click on **Text** shown below.
- Then delete the code which is there and type the code as given below.

Coding:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="20dp">
    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp">
    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
```

```

android:layout_height="wrap_content"
android:layout_weight="1"
android:inputType="numberDecimal"
android:textSize="20sp" />
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:inputType="numberDecimal"
    android:textSize="20sp" />
</LinearLayout>
<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp">
    <Button
        android:id="@+id/Add"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="+"
        android:textSize="30sp"/>
    <Button
        android:id="@+id/Sub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="-"
        android:textSize="30sp"/>
    <Button
        android:id="@+id/Mul"
        android:layout_width="match_parent"

```

```

android:layout_height="wrap_content"
android:layout_weight="1"
android:text="*"
android:textSize="30sp"/>
<Button
android:id="@+id/Div"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
    android:text="/"
    android:textSize="30sp"/>
</LinearLayout>
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:text="Answer is"
    android:textSize="30sp"
    android:gravity="center"/>
</LinearLayout>

```

Java Coding for the Android Application:

- Click on **app** -> **java** -> **com.example. calcapp** -> **MainActivity**.
- Then delete the code which is there and type the code as given below.

Coding

```

package com.example.cse.calcapp;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

```

```

import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity implements view.OnClickListener
{
//Defining the Views
EditText Num1;
EditText Num2;
Button Add;
Button Sub;
Button Mul;
Button Div;
TextView Result;
@Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
//Referring the Views
Num1 = (EditText) findViewById(R.id.editText1);
Num2 = (EditText) findViewById(R.id.editText2);
Add = (Button) findViewById(R.id.Add);
Sub = (Button) findViewById(R.id.Sub);
Mul = (Button) findViewById(R.id.Mul);
Div = (Button) findViewById(R.id.Div);
Result = (TextView) findViewById(R.id.textView);
// set a listener
Add.setOnClickListener(this);
Sub.setOnClickListener(this);
Mul.setOnClickListener(this);
Div.setOnClickListener(this);
}
@Override
public void onClick (View v)
{
float num1 = 0;

```

```

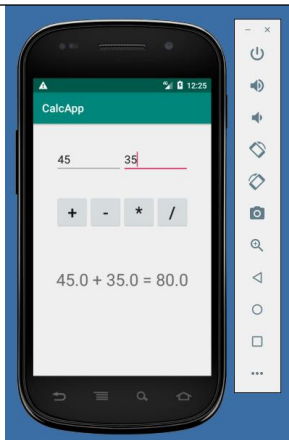
float num2 = 0;
float result = 0;
String oper = "";
// check if the fields are empty
if (TextUtils.isEmpty(Num1.getText().toString()) ||
TextUtils.isEmpty(Num2.getText().toString()))
return;
// read EditText and fill variables with numbers
num1 = Float.parseFloat(Num1.getText().toString());
num2 = Float.parseFloat(Num2.getText().toString());
// defines the button that has been clicked and performs the corresponding operation
// write operation into oper, we will use it later for output
switch (v.getId())
{
case R.id.Add:
    oper = "+";
    result = num1 + num2;
    break;
case R.id.Sub:
    oper = "-";
    result = num1 - num2;
    break;
case R.id.Mul:
    oper = "*";
    result = num1 * num2;
    break;
case R.id.Div:
    oper = "/";
    result = num1 / num2;
    break;
default:
    break;
}
// form the output line

```

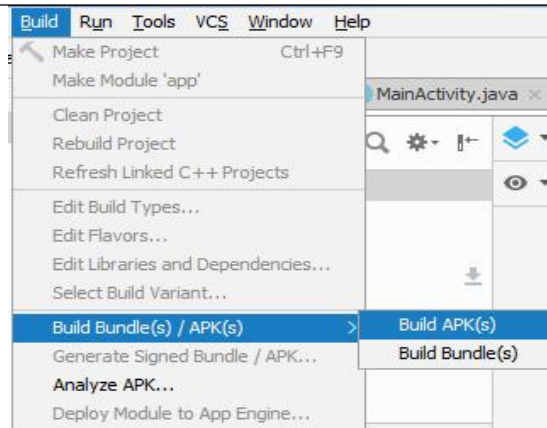
```
Result.setText(num1 + " " + oper + " " + num2 + " = " + result);
}
}
```

Output:

Emulator will be loaded and displays the output of the app developed as shown below.



Select Build APK(s) from Build Menu and build the .apk file for this application. Locate the apk file created and copy it to the mobile Phone to install it and run it from it for verification.



Result:

Thus a Simple Android Application to design a Simple Calculator App has been developed and executed successfully in emulator and Mobile device.

Exercise 8

Implement an Android application that can display different shapes such as circles, rectangles, and triangles on the screen. The application should allow users to select a shape from a dropdown menu and then show the selected shape on the screen.

Aim:

To develop a Simple Android Application to display different shapes.

Procedure:

Creating a New project:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “Shapes”, change the project location and click Next.
- Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Finally click Finish.

Designing layout for the Android Application:

- Click on app -> res -> layout -> activity_main.xml.
- Delete the text there and type the below xml code.

Coding:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
<ImageView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/imageView" />
</RelativeLayout>
```

- Now click on MainActivity.java shown below.
- Delete the system generated java code and type the below code

coding:

```
package com.example.cse.shapes;
```

```

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.widget.ImageView;
public class MainActivity extends Activity
{
@Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
//Creating a Bitmap
Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);
//Setting the Bitmap as background for the ImageView
ImageView i = (ImageView) findViewById(R.id.imageView);
i.setBackgroundDrawable(new BitmapDrawable(bg));
//Creating the Canvas Object
Canvas canvas = new Canvas(bg);
//Creating the Paint Object and set its color & TextSize
Paint paint = new Paint();
paint.setColor(Color.BLUE);
paint.setTextSize(50);
//To draw a Rectangle
canvas.drawText("Rectangle", 420, 150, paint);
canvas.drawRect(400, 200, 650, 700, paint);
paint.setColor(Color.GREEN);
//To draw a Circle
canvas.drawText("Circle", 120, 150, paint);
canvas.drawCircle(200, 350, 150, paint);
paint.setColor(Color.MAGENTA);

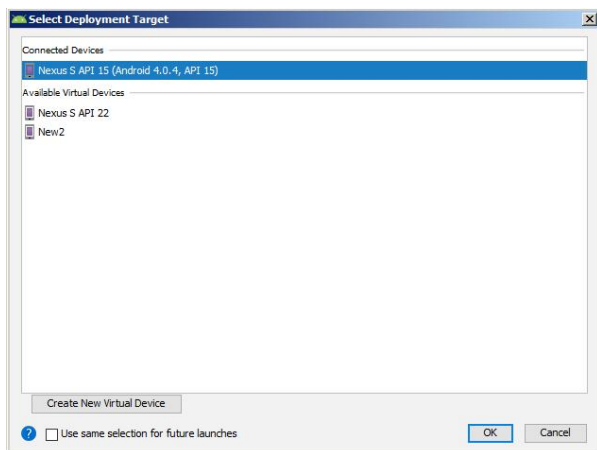
```

```
//To draw a Square
canvas.drawText("Square", 120, 800, paint);
canvas.drawRect(50, 850, 350, 1150, paint);
paint.setColor(Color.BLACK);

//To draw a Line
canvas.drawText("Line", 480, 800, paint);
canvas.drawLine(520, 850, 520, 1150, paint);
}
}
```

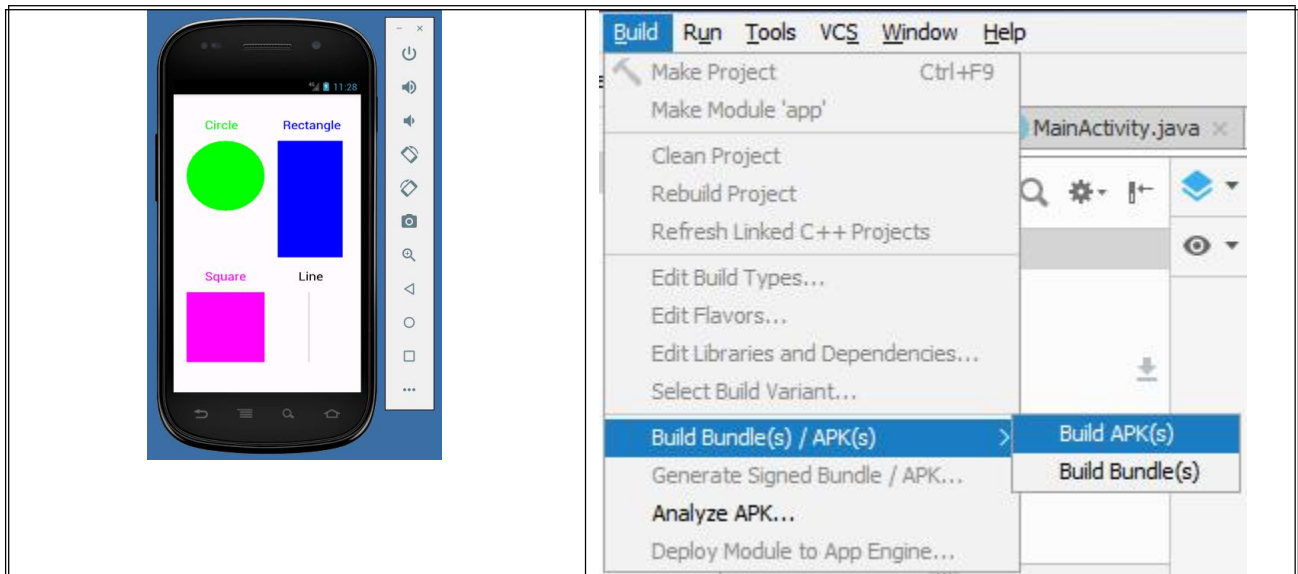
Output:

Click Play icon or press Shift+F10 and select any of the Available Virtual devices from the list or create a New AVD.



Emulator will be loaded and display the output of the app developed as show below.

Select Build APK(s) from Build Menu and build the .apk file for this application. Locate the apk file created and copy it to the mobile Phone to install it and run it from it for verification.



Result:

Thus a Simple Android Application that displays different shapes has been developed and executed successfully in emulator and Mobile device.

Exercise 9

Implement an Android application that demonstrates multi-threading by performing tasks concurrently in separate threads. The application should include a button that triggers the execution of multiple tasks concurrently using different threads.

Aim:

To develop an Android Application that implements Multi threading.

Procedure:

Creating a New project:

Open Android Studio and then click on File -> New -> New project.

Then type the Application name as “ex.no.7” and click Next.

Then select the Minimum SDK as shown below and click Next.

Then select the Empty Activity and click Next.

Finally click Finish.

Designing layout for the Android Application:

Click on app -> res -> layout -> activity_main.xml

Then delete the code which is there and type the code as given below.

Coding:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

<ImageView
    android:id="@+id/imageView"
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:layout_margin="50dp"
    android:layout_gravity="center" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:layout_gravity="center"
    android:text="Load Image 1" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:layout_gravity="center"
    android:text="Load image 2" />
```

```
</LinearLayout>
```

Code for MainActivity.java:

```
package com.example.exno7;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity
{
    ImageView img;
    Button bt1, bt2;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```

bt1 = (Button)findViewById(R.id.button);
bt2= (Button) findViewById(R.id.button2);
img = (ImageView)findViewById(R.id.imageView);

bt1.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        new Thread(new Runnable()
        {
            @Override
            public void run()
            {
                img.post(new Runnable()
                {
                    @Override
                    public void run()
                    {
                        img.setImageResource(R.drawable.india1);
                    }
                });
            }
        }).start();
    }
});

bt2.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        new Thread(new Runnable()
        {

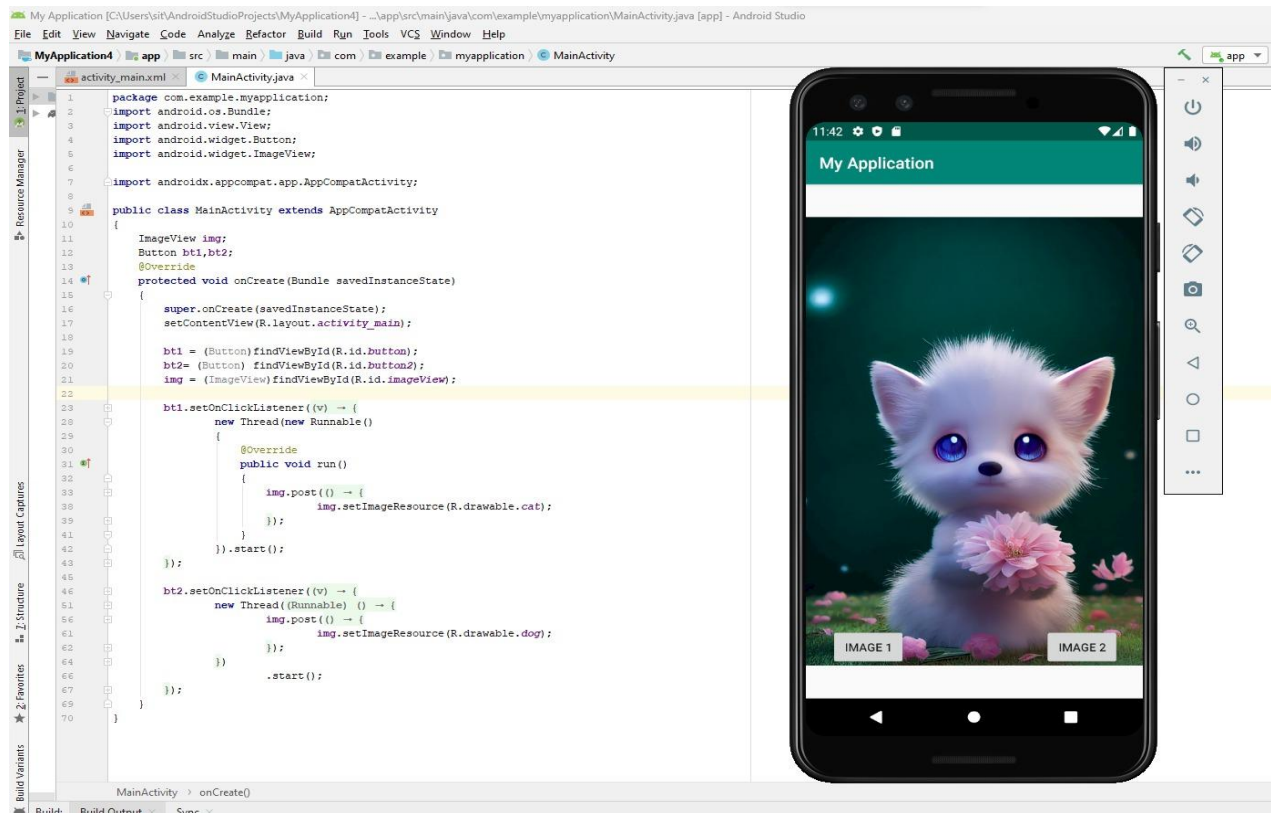
```

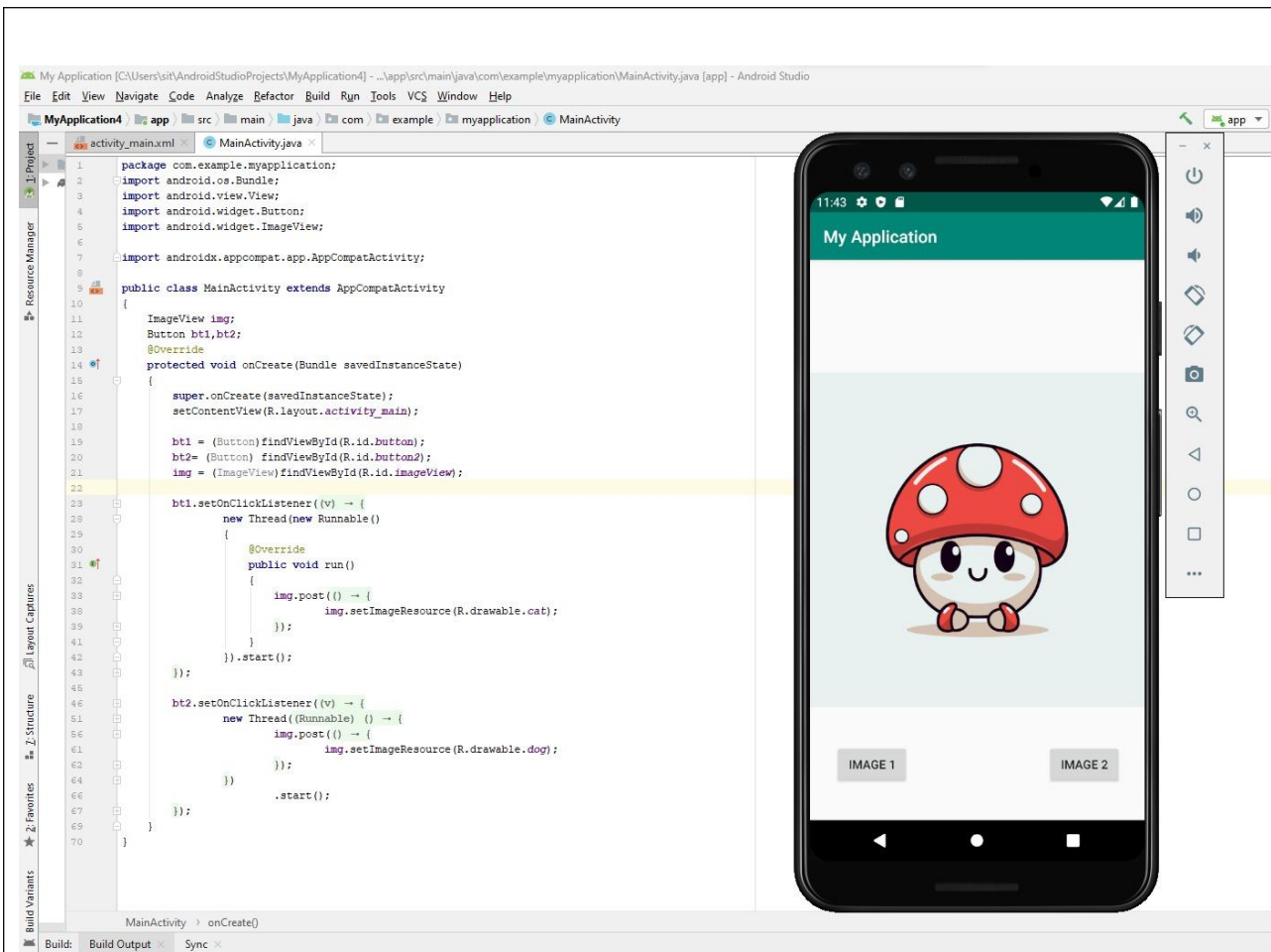
```

@Override
public void run()
{
    img.post(new Runnable()
    {
        @Override
        public void run()
        {
            img.setImageResource(R.drawable.india2);
        }
    });
}
})
.start();
}
});
}
}
}

```

Output:





Result:

Thus an Android Application that implements Multi threading has been developed and executed successfully in emulator and Mobile device.

Exercise 10

Implement an Android application that allows users to create and display notifications on their devices. The application should provide a user interface where users can input notification details such as title, content, and icon, and then show the created notification when a button is clicked.

Aim:

To develop a Simple Android Application to create and display the Notification

Procedure:

Creating a New project:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “Notification”, change the project location and click Next.
- Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Create the Second Activity with the name “SecondAcitivity”
- Click on File -> New -> Activity -> Empty Activity.
- Finally click Finish.

Designing layout for the Android Application:

- Click on app -> res -> layout -> activity_main.xml.
- Now click on Text as shown below.
- Then delete the code which is there and type the code as given below.

Coding:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical">
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Message" android:textSize="30sp" />
    <EditText android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

android:singleLine="true"
android:textSize="30sp" />
<Button android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="30dp"
android:layout_gravity="center"
android:text="Notify"
android:textSize="30sp"/>
</LinearLayout>

```

Designing Layout for Second Activity:

- Click on app -> res -> layout -> activity_second.xml.
- Then delete the code which is there and type the code as given below.

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SecondActivity">
</android.support.constraint.ConstraintLayout>

```

Java Coding for Main Activity

```

package com.example.cse.notification;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity

```

```

{
Button notify;
EditText e;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
notify= (Button) findViewById(R.id.button);
e= (EditText) findViewById(R.id.editText);
notify.setOnClickListener(new View.OnClickListener()
{ @Override
public void onClick(View v)
{
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_launcher).setConten
tIntent(pending).build();
NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE); noti.flags |=
Notification.FLAG_AUTO_CANCEL; manager.notify(0, noti);
}
});
}
}

Java Coding for Second Activity:
package com.example.cse.notification;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; public class SecondActivity extends AppCompatActivity
{
@Override
protected void onCreate(Bundle savedInstanceState)
{

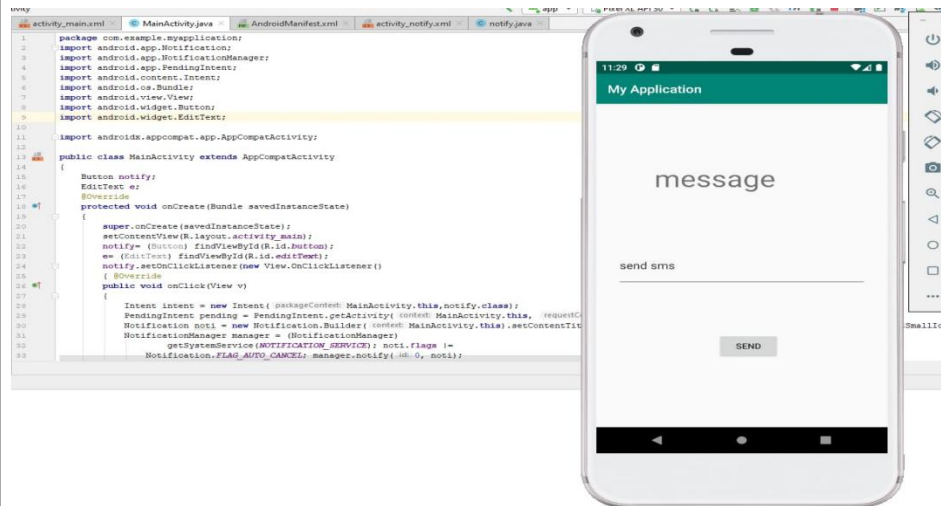
```

```

super.onCreate(savedInstanceState); setContentView(R.layout.activity_second);
}
}

```

Output:



Result:

Thus a Simple Android Application to display the notification has been developed and executed successfully in emulator and Mobile device.

Exercise 11

Implement an Android application that allows users to set an alarm and trigger a notification at the specified time. The application should provide a user interface where users can input the alarm time and set it to activate. When the alarm time is reached, the application should display a notification to alert the user.

Aim:

To develop a Simple Android Application to creating an Alarm

Procedure:

Creating a New project:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “Notification”, change the project location and click Next.
- Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Create the Second Activity with the name “activity_alarm_receiver”
- Click on File -> New -> Activity -> Empty Activity.
- Finally click Finish.

Designing layout for the Android Application:

Coding:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />
    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

android:layout_gravity="center"
android:layout_margin="20dp"
android:checked="false"
android:onClick="OnToggleClicked" />
</LinearLayout>

```

Designing Layout for Second Activity:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AlarmReceiver">
</android.support.constraint.ConstraintLayout>

```

Java Coding for the Android Application:

```

package com.example.cse.alarm;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;
import java.util.Calendar;
public class MainActivity extends AppCompatActivity
{
    TimePicker alarmTimePicker;
    PendingIntent pendingIntent;
    AlarmManager alarmManager;
    @Override
    protected void onCreate(Bundle savedInstanceState)

```

```

{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);
alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
}

public void OnToggleClicked(View view)
{
long time;
if (((ToggleButton) view).isChecked())
{
Toast.makeText(MainActivity.this, "ALARM ON", Toast.LENGTH_SHORT).show();
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour();
calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute();
Intent intent = new Intent(this, AlarmReceiver.class);
pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);
time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));
if(System.currentTimeMillis()>time)
{
if (calendar.AM_PM == 0)
time = time + (1000*60*60*12);
else
time = time + (1000*60*60*24);
}
alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000, pendingIntent);
}
else
{
alarmManager.cancel(pendingIntent);
Toast.makeText(MainActivity.this, "ALARM OFF", Toast.LENGTH_SHORT).show();
}
}
}

```


Java Coding for Second Activity:

```
package com.example.cse.alarm;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.widget.Toast;
public class AlarmReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        Toast.makeText(context, "Alarm! Wake up! Wake up!", Toast.LENGTH_LONG).show();
        Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        if (alarmUri == null)
        {
            alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        }
        Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
        ringtone.play();
    }
}
```

XML Coding for Android Manifest file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.cse.alarm" >
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportRtl="true"
android:theme="@style/AppTheme" >
```

```

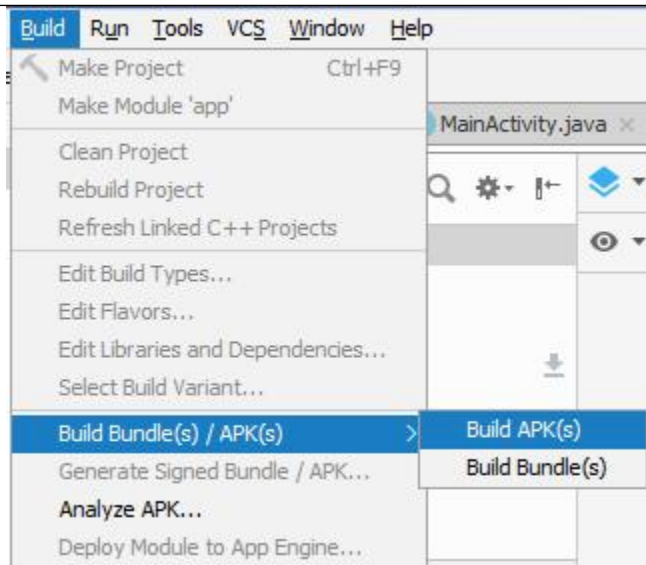
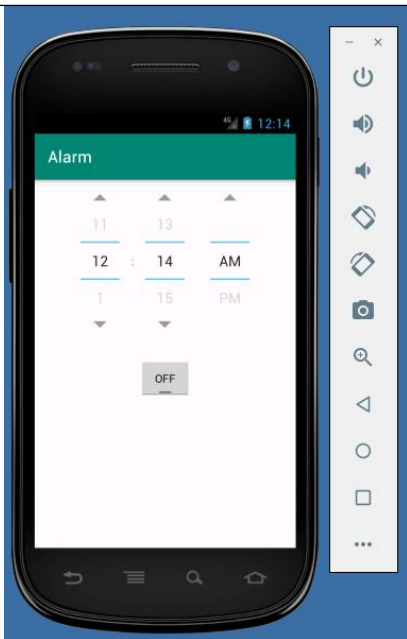
<activity android:name=".MainActivity" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name=".AlarmReceiver" >
</receiver>
</application>
</manifest>

```

Output:

Emulator will be loaded and displays the output of the app developed as shown below. (If the analog clock is not supported, then the digital clock will be displayed.)

Select Build APK(s) from Build Menu and build the .apk file for this application. Locate the apk file created and copy it to the mobile Phone to install it and run it from it for verification.



Result:

Thus a Simple Android Application to simulate the alarm has been developed and executed successfully in emulator and Mobile device.

Exercise 12

Implement an Android application that calculates the Body Mass Index (BMI) based on the user's weight and height. The application should provide a user interface where users can input their weight and height, and upon clicking a button, the BMI value should be calculated and displayed.

Aim:

To develop a Simple Android Application to design a Simple BMI Calculator

Procedure:

Creating a New project:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “BMICalc”, change the project location and click Next.
- Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Finally click Finish.

Designing layout for the Android Application:

Coding:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/images1"
    android:fadingEdge="horizontal"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/tv1"
        android:layout_width="124dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:paddingLeft="15dp"
        android:paddingTop="40dp"
        android:shadowColor="@android:color/black"
        android:shadowDx="4"
```

```

android:shadowDy="4"
android:text="BMI"
android:textAppearance="?android:attr/textAppearanceLarge"
android:textColor="@android:color/white"
android:textSize="50sp"
android:typeface="serif" />
<TextView android:id="@+id/tv2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:text="Calculator"
android:textColor="@android:color/white"
android:textSize="20dp"
android:textStyle="bold" />
<TextView android:id="@+id/tv3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:paddingTop="30dp"
android:text="WEIGHT (KG)"
android:textAppearance="?android:attr/textAppearanceMedium"
android:textColor="@android:color/white"
android:textStyle="bold|italic"
android:typeface="serif" />
<EditText android:id="@+id/et1"
android:layout_width="96dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:hint="IN KGs"
android:ems="10"
android:fadingEdgeLength="10dp"
android:inputType="numberDecimal"
android:textAlignment="center" >
<requestFocus />

```

```

</EditText>
<TextView android:id="@+id/tv4"
    android:layout_width="151dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:foregroundGravity="center_horizontal"
    android:gravity="center_horizontal"
    android:paddingTop="30dp"
    android:text="HEIGHT (CM)"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="@android:color/white"
    android:textStyle="bold|italic"
    android:typeface="serif" />
<EditText android:id="@+id/et2"
    android:layout_width="96dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:hint="IN CMs"
    android:ems="10"
    android:inputType="numberDecimal" >
</EditText>
<Button
    android:id="@+id/ib1"
    android:layout_width="158dp"
    android:layout_height="51dp"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:fadingEdge="vertical"
    android:longClickable="true"
    android:nextFocusRight="@android:color/holo_orange_dark"
    android:text="Calculate"
    android:visibility="visible" />
<TextView
    android:id="@+id/tv5"

```

```

android:layout_width="278dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:gravity="center"
android:paddingTop="20dp"
android:text=""
android:textColor="@android:color/holo_orange_dark"
android:textSize="20dp"
android:textStyle="bold" />
</LinearLayout>

```

Java Coding for the Android Application:

```

package akn.bmicale;

//Import necessary package and file
import android.os.Bundle;
import android.app.Activity;
import android.text.TextUtils;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

//Main activity class start here
public class MainActivity extends Activity
{
    //Define layout
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Get the references to the widgets
        final EditText e1 = (EditText) findViewById(R.id.et1);
        final EditText e2 = (EditText) findViewById(R.id.et2);
        final TextView tv5 = (TextView) findViewById(R.id.tv5);
        findViewById(R.id.ib1).setOnClickListener(new View.OnClickListener()

```

```

{
// Logic for validation, input can't be empty
@Override
public void onClick(View v)
{
String str1 = e1.getText().toString();
String str2 = e2.getText().toString(); if(TextUtils.isEmpty(str1))
{
e1.setError("Please enter your weight");
e1.requestFocus(); return;
}
if(TextUtils.isEmpty(str2))
{
e2.setError("Please enter your height");
e2.requestFocus();
return;
}
//Get the user values from the widget reference
float weight = Float.parseFloat(str1);
float height = Float.parseFloat(str2)/100;
//Calculate BMI value
float bmiValue = calculateBMI(weight, height);
//Define the meaning of the bmi value
String bmiInterpretation = interpretBMI(bmiValue);
tv5.setText(String.valueOf(bmiValue + " - " + bmiInterpretation));
}
});
}
//Calculate BMI
private float calculateBMI (float weight, float height)
{
return (float) Math.round((weight / (height * height))*100)/100;
}
// Interpret what BMI means

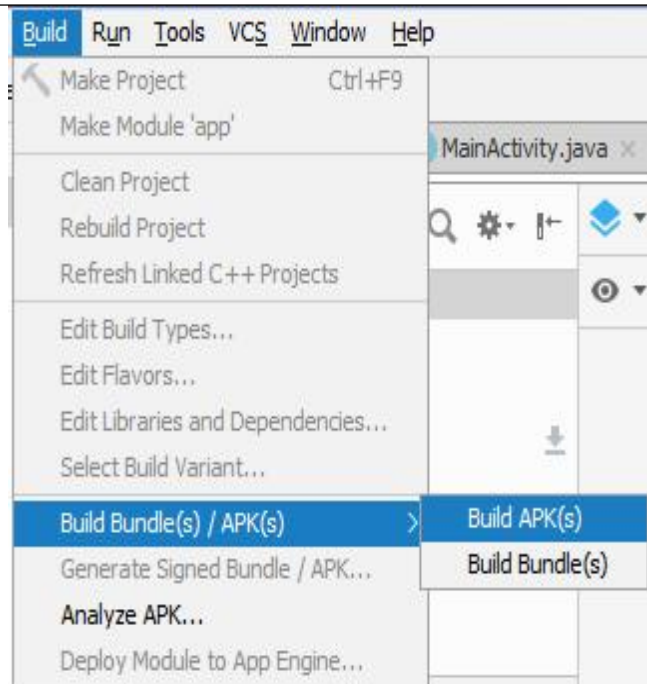
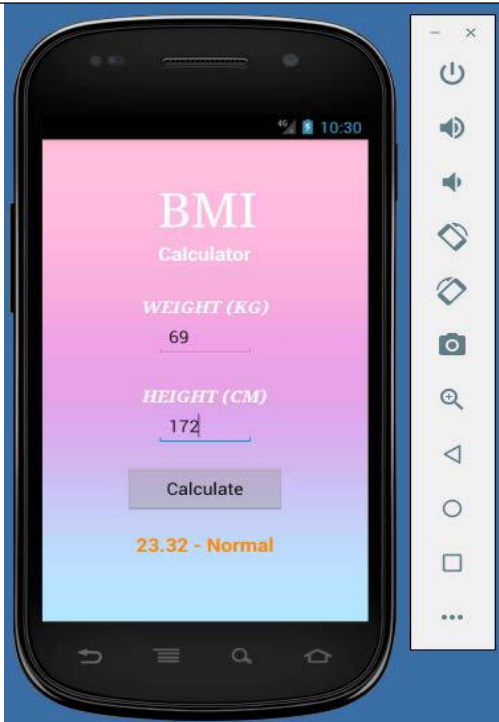
```

```
private String interpretBMI(float bmiValue)
{
    if (bmiValue < 16)
    {
        return "Severely underweight";
    }
    else if (bmiValue < 18.5)
    {
        return "Underweight";
    }
    else if (bmiValue < 25)
    {
        return "Normal";
    }
    else if (bmiValue < 30)
    {
        return "Overweight";
    }
    else
    {
        return "Obese";
    }
}
}
```


Output:

Emulator will be loaded and displays the output of the app developed as shown below.

Select Build APK(s) from Build Menu and build the .apk file for this application. Locate the apk file created and copy it to the mobile Phone to install it and run it from it for verification.

**Result:**

Thus a Simple Android Application to calculate the Body Mass Index (BMI) has been developed and executed successfully in emulator and Mobile device.