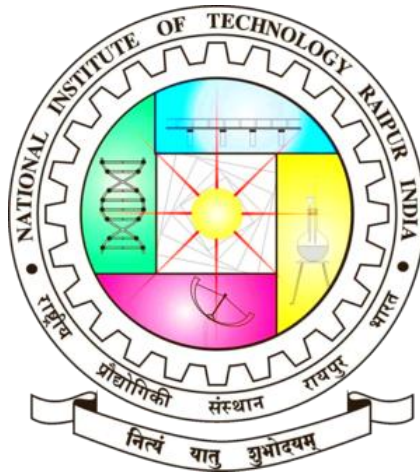


National Institute of Technology, Raipur



Computer Graphics Term Project

Name: Ayush Ganguli

Branch: Information Technology

Semester: 5th

Roll No.: 18118013

Submitted To: Dr. Sanjay Kumar

| <u>S.No.</u> | <u>Project Name</u> | <u>Pages</u> |
|--------------|---|--------------|
| <u>1</u> | <u>Program to implement digital clock</u> | <u>3-6</u> |
| <u>2</u> | <u>Program to implement a rainbow</u> | <u>7-10</u> |
| <u>3</u> | <u>Program to implement a moving car</u> | <u>10-16</u> |

INDEX

1.Program to Implement a Digital Clock

- **Tech-Stack Used:** JavaScript, HyperText Markup Language(HTML 5), Cascading Style Sheets(CSS 3),setTimeout() inbuilt javaScript Function
- **Aim:** The project aims to implement a digital clock that displays the time in Hours Minutes Seconds XM(AM/PM) 12-hour format along with the day of the week, day of the month, month name, year. Real-time data is shown.
- **Working:**
 - The project works based on repeatedly creating a current date-time object through setTimeout() function recursively and repeatedly manipulating the Document Object Model(DOM) at set intervals to display the current time.
 - **setTimeout():** It allows us to run a function once after an interval of time.
 - Here the same function is called recursively after a fixed interval of time.
- **Procedure:**
 - A user interface and design are created using HTML and CSS. Styling including font type, font color, background color, text-alignment is done using CSS.
 - As soon as the page loads an onload event showTime() is called.
 - showtime() creates a date-time object which contains the current date-time.
 - Time is stored in 24 hours format and it is converted to 12 hr format.
 - Day of week name, month name is stored as integer i.e 0 for Sunday 1 for Monday and so on. They are converted into words by mapping every integer index of the array to a day of week name and month name.
 - JavaScript is used for DOM manipulation to change the inner HTML content of the time field. Date time is successfully overwritten in the time field.
 - After a small interval showTime is called again and the procedure repeats. This process goes on infinitely
- **Code:**

- **HTML Code(index.html)**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Digital Clock</title>
</head>
<link rel="stylesheet" href="style.css">
<script src="script.js"></script>
<body onload="showTime()">
  <div id="date_time_container">
    <div class="wrapper">
      <div id="time"></div>
      <div id="date"></div>
    </div>
  </div>
</body>
</html>
```

- **CSS Code(style.css)**

```

• body
• {
•     background-color: #333;
•     overflow-x: hidden;
• }
• .wrapper {
•     padding:25px;
•     max-width:1500px;
•     width:100%;
•     text-align:center;
•     margin:0 auto;
•
• }
• #time{
•     background-color:#333;
•     font-family: sans-serif;
•     font-size:200px;
•     color:#fff;
• }
• #time span {
•     color:#888;
•     font-size:300px;
•     position:relative;
•     top:-5px;
•     left:10px;
• }
• #date {
•     letter-spacing:10px;
•     font-size:80px;
•     font-family:arial,sans-serif;
•     color:#fff;
• }

```

- JavaScript Code(script.js)

Please Turn Over to next Page

```

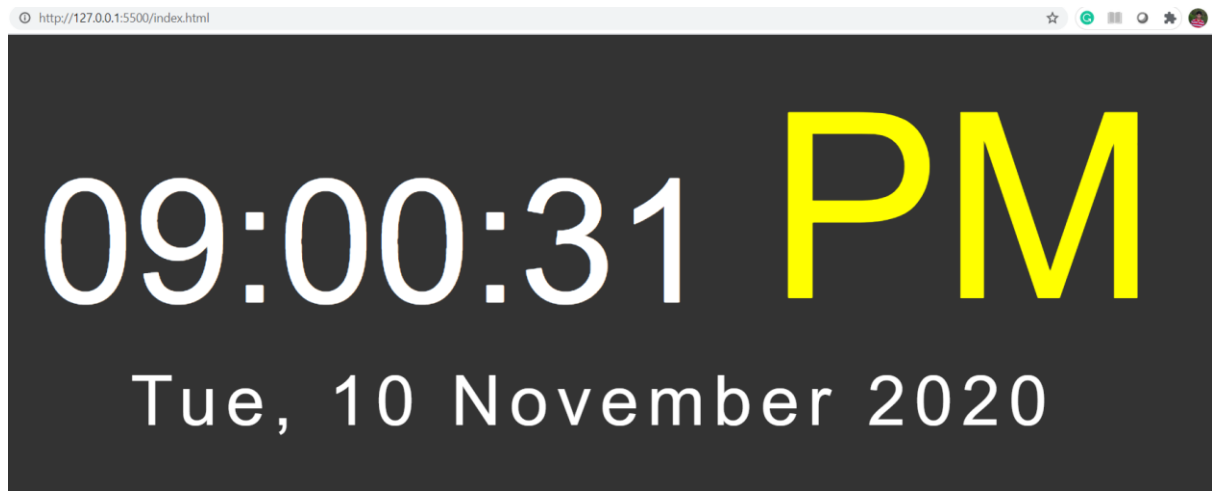
function addZero(time_input) {
    if (time_input < 10) {
        time_input = "0" + time_input;
    }
    return time_input;
}

function showTime() {
    var curr_time = new Date();
    var hr = curr_time.getHours();
    var min = curr_time.getMinutes();
    var sec = curr_time.getSeconds();
    var suffix_time;
    if (hr < 12)
    {
        suffix_time = "<span style='color:yellow;'>AM</span>";
    }
    else
    {
        suffix_time = "<span style='color:yellow;'>PM</span>";
    }
    if (hr == 0)
    {
        hr = 12;
    }
    if (hr > 12)
    {
        hr -= 12;
    }
    // adding 0 to all time field less than 10

    hr = addZero(hr);
    min = addZero(min);
    sec = addZero(sec);
    document.getElementById("time").innerHTML = hr + ":" + min + ":" + sec + "
" + suffix_time;
    var day_name_arr = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];
    var month_arr = ['January', 'February', 'March', 'April', 'May', 'June', '
July', 'August', 'September', 'October', 'November', 'December'];
    var curWeekDay = day_name_arr[curr_time.getDay()];
    var curDay = curr_time.getDate();
    var curMonth = month_arr[curr_time.getMonth()];
    var curYear = curr_time.getFullYear();
    var date = curWeekDay+", "+curDay+" "+curMonth+" "+curYear;
    document.getElementById("date").innerHTML = date;
    setTimeout(function(){ showTime() }, 500);
}

```

- Result: A digital clock has successfully been created.



2.Program to implement a rainbow

- Tech Stack Used: Python(3.8.2), Turtle(python standard graphics library)
- Aim: The program aims to create a rainbow consisting of all 7 colors or a custom rainbow consisting of only specific colors based on user input
- Working:

The program uses turtle a python standard graphic library. Turtle is like a drawing board that let us draw all over a screen through a few standard functions.

This program uses turtle to draw concentric circles of different colors. The program gives users a choice to draw a rainbow of all 7 colors or draw a rainbow with specific colors only.

- How to run the program?
 - Navigate to the directory containing the rainbow_main.py file using terminal
 - Type the following command to start the program(python version above 3.5 required):

`python rainbow_main.py`
 - Now press 1 to generate a rainbow consisting of all 7 colors.
 - Press any key other than 1 to generate a custom rainbow with specific colors with colors as listed by the program. In the next line enter the color numbers you want in your rainbow as space-separated input.
 - A turtle window pops up and a rainbow is generated according to user choice.

- Code
Please turn over for code

```

import turtle
import time
def drawSemiCircleFunction(col,rad,val):
    my_turtle.color(col)
    my_turtle.circle(rad,-180)
    my_turtle.up()
    my_turtle.setpos(val,0)
    my_turtle.down()
    my_turtle.right(180)

colours = ['violet','indigo','blue','green','yellow','orange','red']
print("If you want rainbow with all colours press 1 else if you want partial rainbow press any other key")
bool_val = int(input())

if bool_val == 1:
    col = colours
else:
    print("You have chosen the option to make a custom rainbow.")
    print("Choose the colours from list below. Enter the colours you want to choose as space separated input")
    for i in range(len(colours)):
        print("%d -- %s"%(i+1,colours[i]))
    colours_index = list(map(int,input().strip().split()))
    col = []
    for x in colours_index:
        col.append(colours[x-1])
my_turtle = turtle.Turtle()
screen = turtle.Screen()
screen.setup(600,600)
screen.bgcolor('black')
my_turtle.right(90)
my_turtle.width(10)
my_turtle.speed(7)
for i in range(len(col)):
    drawSemiCircleFunction(col[i],10*(i+8),-10*(i+1))
time.sleep(10)
my_turtle.hideturtle()

```

Result:

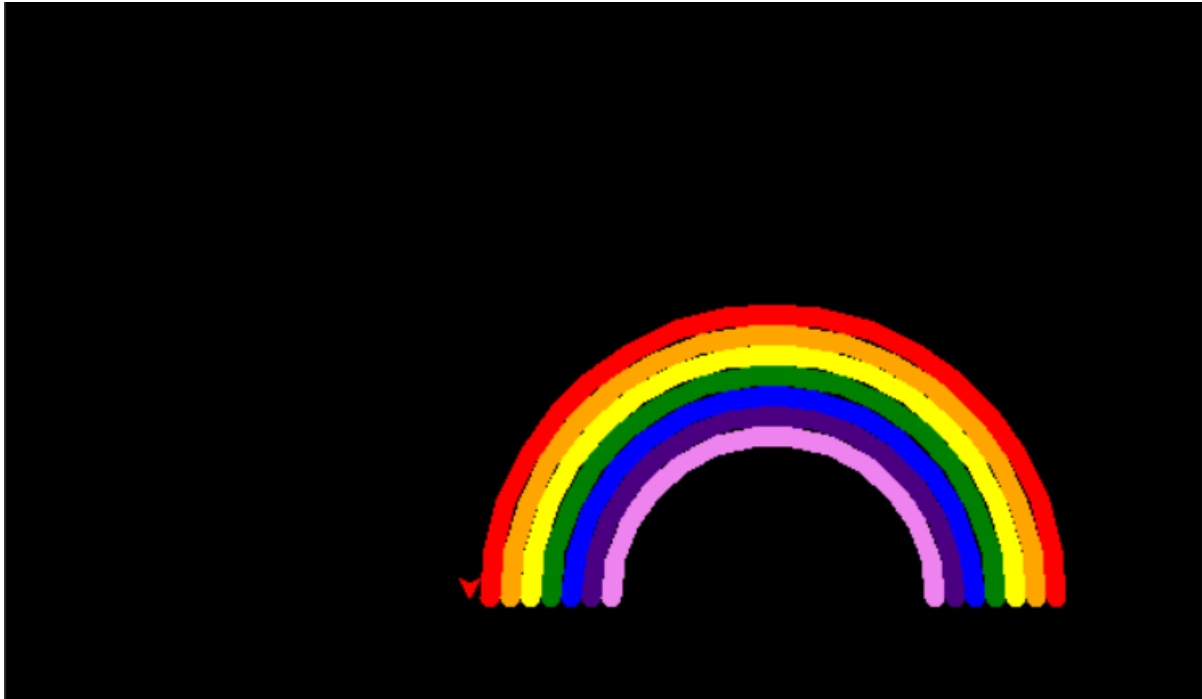
Rainbow with all 7 colors and custom rainbow with specific colors has been created.

Rainbow with all 7 colors

```

C:\Users\Ayush Ganguli\Desktop\CG Project\rainbow>python rainbow_main.py
If you want rainbow with all colours press 1 else if you want partial rainbow press any other key
1

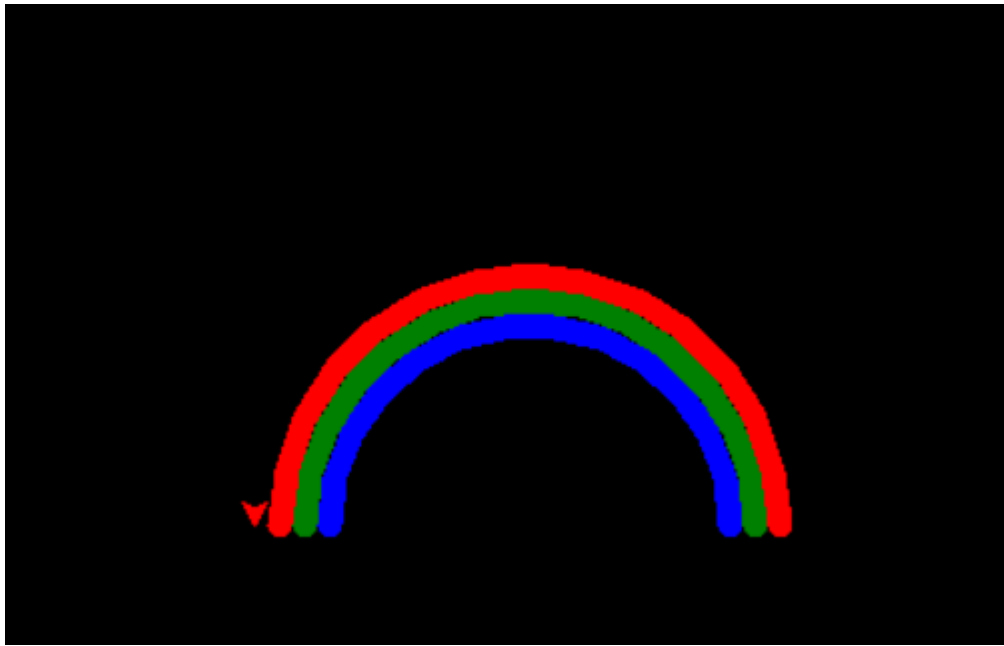
```

Blue Green Red Rainbow(Partial Rainbow)

```
C:\Users\Ayush Ganguli\Desktop\CG Project\rainbow>python rainbow_main.py
If you want rainbow with all colours press 1 else if you want partial rainbow press any other key
2
You have choosen the option to make a custom rainbow.
Choose the colours from list below. Enter the colours you want to choose as space seperated input
1 -- violet
2 -- indigo
3 -- blue
4 -- green
5 -- yellow
6 -- orange
7 -- red
3 4 7
```

Please Turn Over



3.Program to Implement Moving Car

- Tech Stack Used: JavaScript, Hyper Text Markup Language(HTML 5), Cascading Style Sheets(CSS 3), CSS animations
- Aim: To create a moving whose movement can be controlled with the help of a start and stop button
- Working:

The car is actually static about its position. The road is rotating like a conveyer belt in the backward direction and the wheels of the car which are a separate image are rotating in a clockwise direction. This gives the viewer an illusion that the car is actually moving. Plus engine sound audio controlled through javaScript also contributes towards making this illusion seem real. The car also shakes further adding a better visual effect.

All the animation is done through CSS keyFrames controlled through javaScript. KeyFrames specify animation rule. Animations are created by gradually changing the styling according to keyFrame rules.

- Procedure:
 - An image of a car was photoshopped. Its body was separated from its 2 wheels.
 - In the top part of the HTML page scenery along with trees was created.
 - In the bottom, a road was set as a background image.
 - The car along with its 2 wheels are positioned on the webpage
 - The overlapping is done in such a way that the car appears on the top layer
 - A hidden Boolean text field containing 0 and 1 and a button is added to the page to control the car.
 - As soon as the button is clicked onclick event gets activated. fn1() function is called. Since the Boolean field is 0, the start car method starts working.
 - The track is animated to move backward and the wheels of the car are rotated in a clockwise direction.
 - The car is moved linearly along the y-axis to give a shaking effect.
 - Engine sound audio starts playing.
 - The boolean field is set to 1. As soon as the stop button is clicked all the shaking, road moving backward, wheels rotating, the audio effect is stopped. The boolean field is again set to 0.

- Code

HTML Code(index.html)

Please Turn Over

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>My Moving Car</title>
  <link rel="stylesheet" href="style.css">
  <script src="jquery-3.5.1.min.js"></script>
  <script src="script.js"></script>
</head>
<body>
  <input type="hidden" value="0" id="bool_check">
  <div class="container">
    <div style="position: absolute ; z-
index: 3;"><a href="#" class="button" onclick="fn1(this)" >START CAR
</a></div>
    <div id="sky">
      <div id="trees"></div>
      <div id="track"></div>
      <div id="car">
        <div class="wheel wheel1">
          
          </div>
          <div class="wheel wheel2">
            
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

CSS Code(style.css)

Please Turn Over

```

*{
    margin: 0;
    padding: 0;
}
.button {
    background-color: #4CAF50; /* Green */
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
}
body{
    overflow: hidden;
}
#sky{
    height: 100vh;
    width: 100%;
    background-image: url(star_backgrounjpg.jpg);
    background-repeat: no-repeat;
    position: absolute;
}

#trees{
    height: 100vh;
    width: 100%;
    background-image: url(trees.png);
    background-size: cover;
    position: absolute;
    top: -144px;
}
#track{
    height: 60vh;
    width: 800vw;
    background-image: url(track.png);
    background-repeat: repeat-x;
    position: absolute;
    top: 70vh;
}

#car{
    height: 100px;
    width: 380px;
    background-image: url(car_body.png);
    background-size: cover;
    background-repeat: no-repeat;
    position: absolute;
    left: 444px;
    bottom: 30vh;
}

#wheel1_img
{
    width: 77px;
    position: relative;
    top: 41.4px;
    left: 41px;
}

```

```

#wheel2_img
{
    width: 77px;
    position: relative;
    top: -38.8px;
    left: 233px;
}
@keyframes rotateWheelFunction
{
    100%{
        transform: rotate(360deg);
    }
}
@keyframes moveCarFunction
{
    100%{
        transform: translateX(-500vw);
    }
}
@keyframes bounceFunction
{
    0%{
        transform: translateY(-5px);
    }
    50%{
        transform: translateY(5px);
    }
    100%{
        transform: translateY(-5px);
    }
}

```

JavaScript Code(script.js)

```

var engine_sound;
function fn1(btn)
{
    bool_check_div = document.getElementById('bool_check');
    bool_check = parseInt(bool_check_div.value);
    track = document.getElementById('track');
    car = document.getElementById('car');
    wheel1_img = document.getElementById('wheel1_img');
    wheel2_img = document.getElementById('wheel2_img');
    if (bool_check == 0)
    {
        track.style.animation = 'moveCarFunction linear 13s infinite';
        car.style.animation = 'bounceFunction linear .3s infinite';
        wheel1_img.style.animation = 'rotateWheelFunction linear 0.6s infinite';
        wheel2_img.style.animation = 'rotateWheelFunction linear 0.6s infinite';
        bool_check_div.value = '1';
        btn.innerHTML = 'Stop Car';
        engine_sound = document.createElement('audio');
        engine_sound.id = 'engine_sound_element';
        engine_sound.setAttribute('src','engine_sound.mp3');
        engine_sound.loop = true;
        engine_sound.play();
    }
    else
    {

```

```

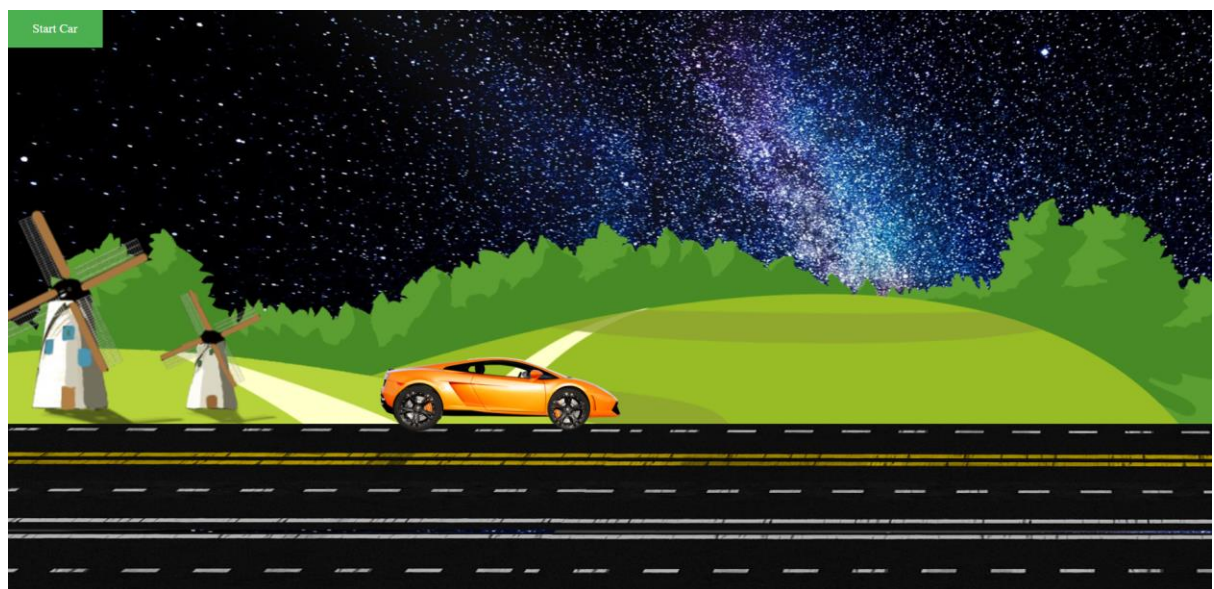
track.style.animation = 'moveCarFunction linear 0.1s 1s';
car.style.animation = 'bounceFunction linear .1s 1s';
wheel1_img.style.animation = 'rotateWheelFunction linear 0.1s 1s';
wheel2_img.style.animation = 'rotateWheelFunction linear 0.1s 1s';
bool_check_div.value = '0';
btn.innerHTML = 'Start Car';
engine_sound.pause();
}
}

```

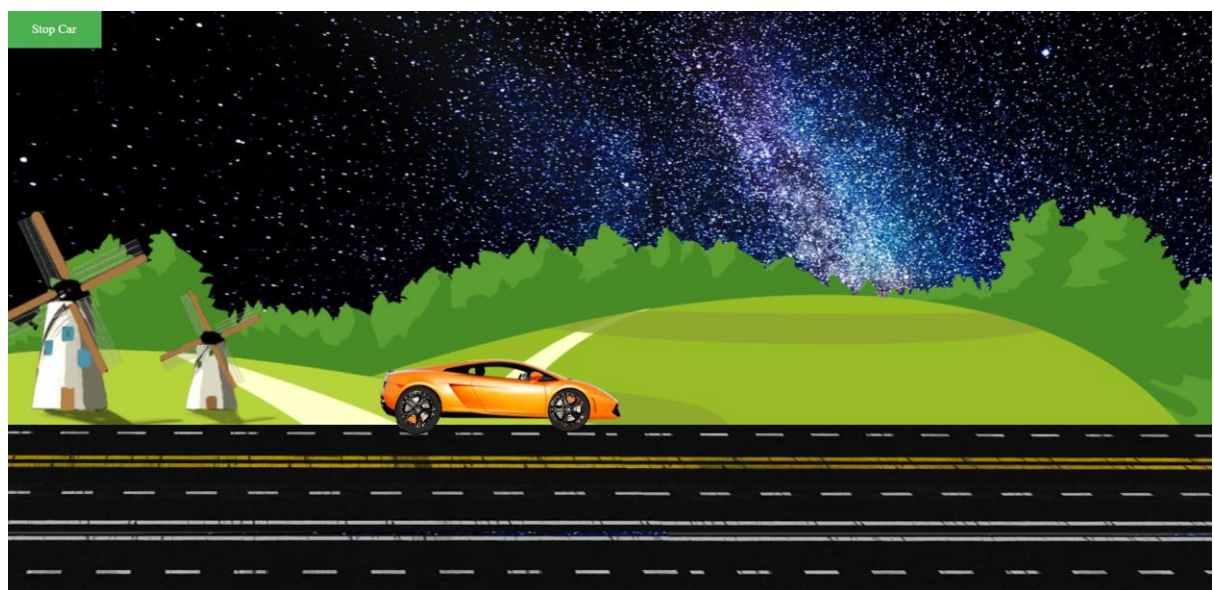
➤ Result

Following are some snaps of the output obtained:

When the car is stationary



When the car is in motion



Please Note: The animation effects can only be experienced by viewing the project.
Please open the index.html file through google chrome of moving_car_final_code folder
uploaded to experience the effects.