Project Report

On

Project Title: Arcade Game

By

Prathmesh Dattatray Kamble Roll no: 103
Devang Shah Roll no: 72

As part of Second Year BBA (CA) Project (CA 405)

AY (2020-2021)
CBCS 2019 Pattern, Savitribai Phule Pune University

Marathwada Mitra Mandal College of Commerce
Deccan Gymkhana, Pune-411004

_____	**Dr. Anpat S. M.**	**Dr. Devidas Golhar**
Guide	**Head of Department**	**Principal**

CERTIFICATE

This is to certify that Miss / Mr.
Prathmesh Dattatray Kamble, Roll No. 103
and Miss / Mr. Devang Shah, Roll No. 72
from SYBBA(CA) have satisfactorily completed the Project titled Arcade Game
during the academic year
2020-2021 as per requirement of the SYBBA(CA) Project (CA-405) of Savitribai Phule University CBCS 2019 Syllabus.

_____	Dr. Anpat S. M.	Dr. Devidas Golhar
Guide	Head of Department	Principal

ACKNOWLEDGEMENT

I hereby declare that the SYBBA(CA) project (CA-405) titled Arcade Game has been submitted to the Department of BBA (CA) Marathwada Mitra Mandal College of Commerce, Deccan Gymkhana, Pune – 411004 for the fourth Semester BBA (CA) degree. It is a record of original work done by me under the supervision and guidance of Komal Dhoot and we are also thankful to all teachers and staff members of our department for their kind cooperation and help.

We are also grateful to Dr Sandip Anpat, Head of Department and our respected Principal, Dr. Devidas Golhar for permitting us to utilize all the necessary services or facilities provided by the college.

Place-Pune.

Date-

(Name and Signature the Student)

INDEX

1. Introduction to Project

Project Name: - Arcade Game

Project Description: -

The space shooter is a very attractive game for any age's peoples.
The spaceship moves around the galaxy and protect galaxy from various asteroids and enemies. Asteroids and enemies attack the spaceship. The spaceship shoots and destroy his enemies and save galaxy. The score is being increased automatically by destroying each asteroids and enemies. The gameplay is an infinite side-scrolling shooter where enemies randomly appear and start firing at you. Your objective is to survive and destroy as much enemies as you can to get points. The spaceship can move in all directions, but it can't go outside the boundary of the screen.

2. Why This Project?

Since the 1970s, people started to take interest in using their computers as an entertainment environment, thus, the multibillion game industry was starting to take shape. Having presented earlier the sum of money this industry produces; I decided to have a go and create a game of my own. As a kid, I was always fascinated by the idea of becoming a game developer, but, as years went by, I have realized this is not exactly what programming and computer science, as a practice, are about and I dropped the idea. However, the second-year project offered me the possibility to try and achieve one of my childhood's dreams and I could not resist the temptation.

3. Objectives

The game is developed for full-time entertainment and enthusiasms. It teaches the Gamer to be alert at every situation he/she faces, because if the Gamer is not fully alert and notice the saucer fire, he/she must be hit by the saucer-bombs.
Though the proposed game is an action game, it does not involve direct violence. No zombie killing, animal killing, or human killing is performed in the game. So, it can also be viewed as a non-violence game.
Kids can also play this game, because the design of the game is very simple, controlling the game is very easy – pressing some neighboring keys of the keyboard.

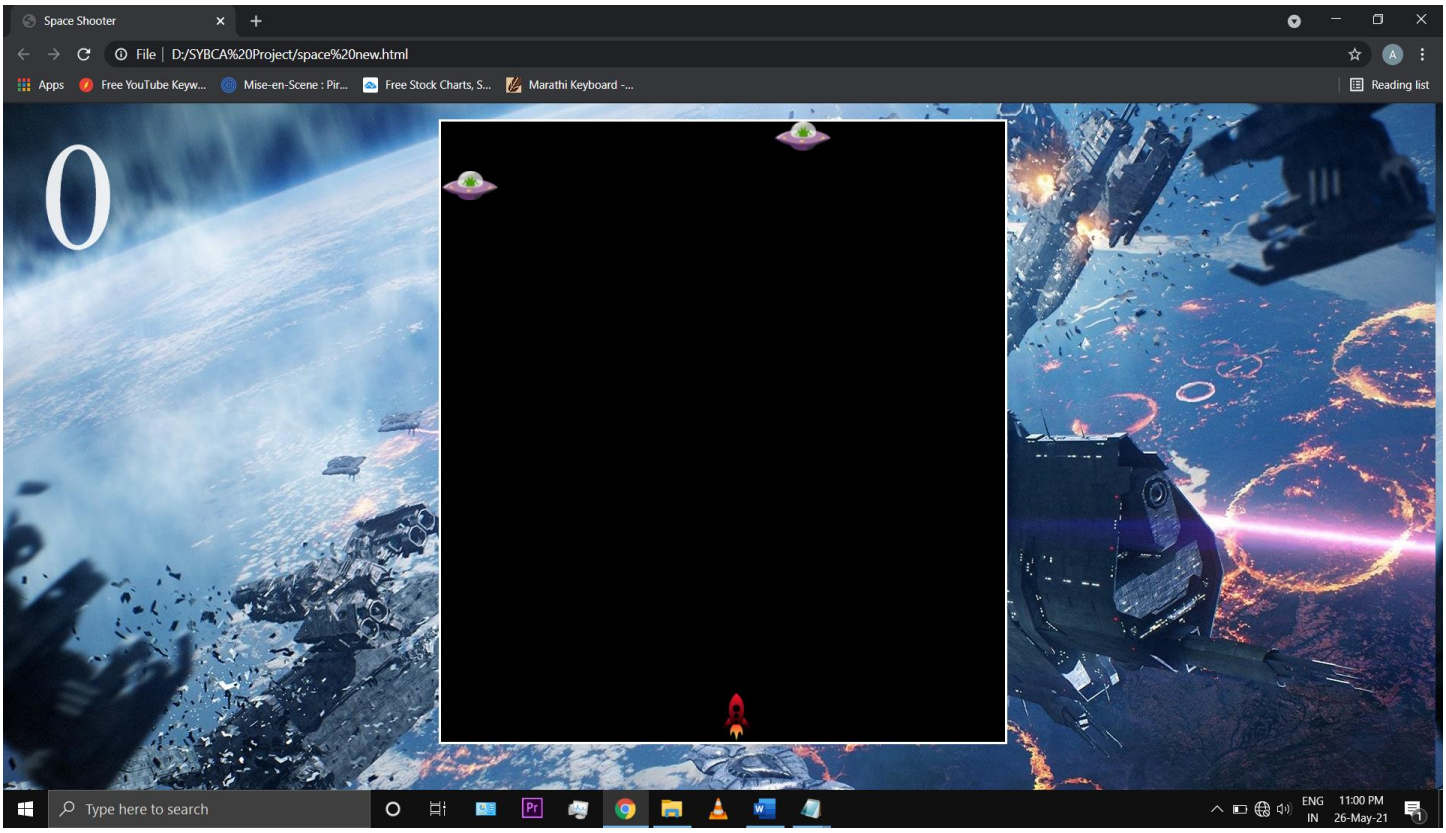
4. Scope of Our Game

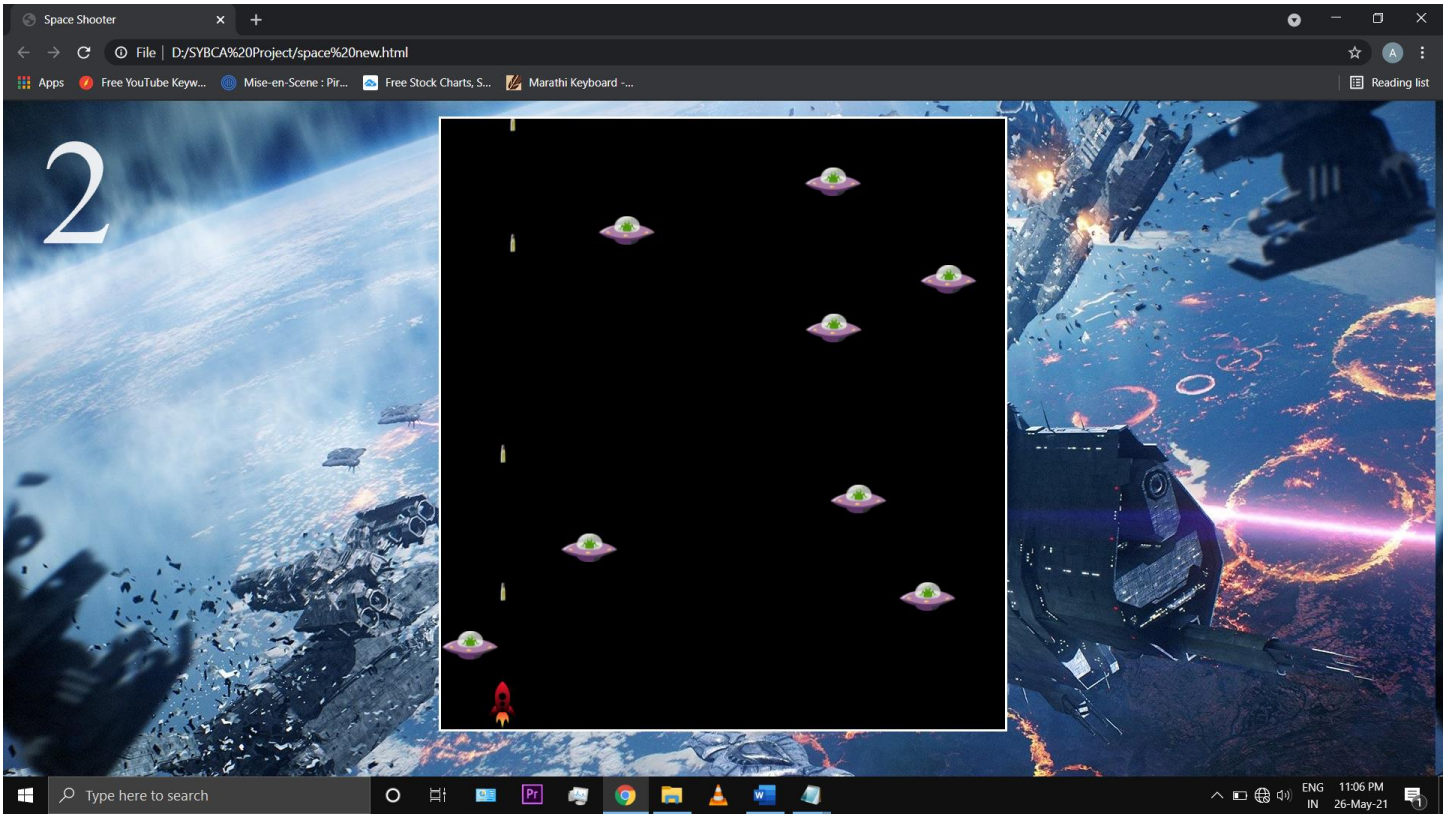
This Report describes all the requirements for the project. The purpose of this research is to provide a virtual image for the combination of both structured and unstructured information of my project "Space Shooter". This is a single-player strategy game on the Windows platform. The player will progress through levels which require precise manipulation of the environment, though the game Encourages creativity and daring via branching pathways. The episodic structure of the game facilitates the pace of the story. I demonstrate the action flow between inputs, script, display (output). We are working mainly with story, levels, object, animation, graphics, scripts, game engine facilities.

5. Features

- Attractive background 
- Spaceship 
- Different asteroid types
- Different enemy types 
- Beautiful graphics
- Live score 
- Background music

6. Screen design layout





7. Planning and Scheduling

Project planning and scheduling is a part of project management. The project planning stage requires several inputs, including conceptual proposals, project schedules. The development of this project is not successfully done without proper planning and scheduling. Project planning and scheduling is very important stage for us.

I. **Analysis:** The maximum time for analysis phase of this project is 2 days.

II. **Design:** The maximum time for design phase of this project is 2 days.

III. **Implementation:** The maximum time for implementation phase of this project is 2 days.

IV. **Testing:** The maximum time for testing phase of this project is 1 days.

8. Development Process

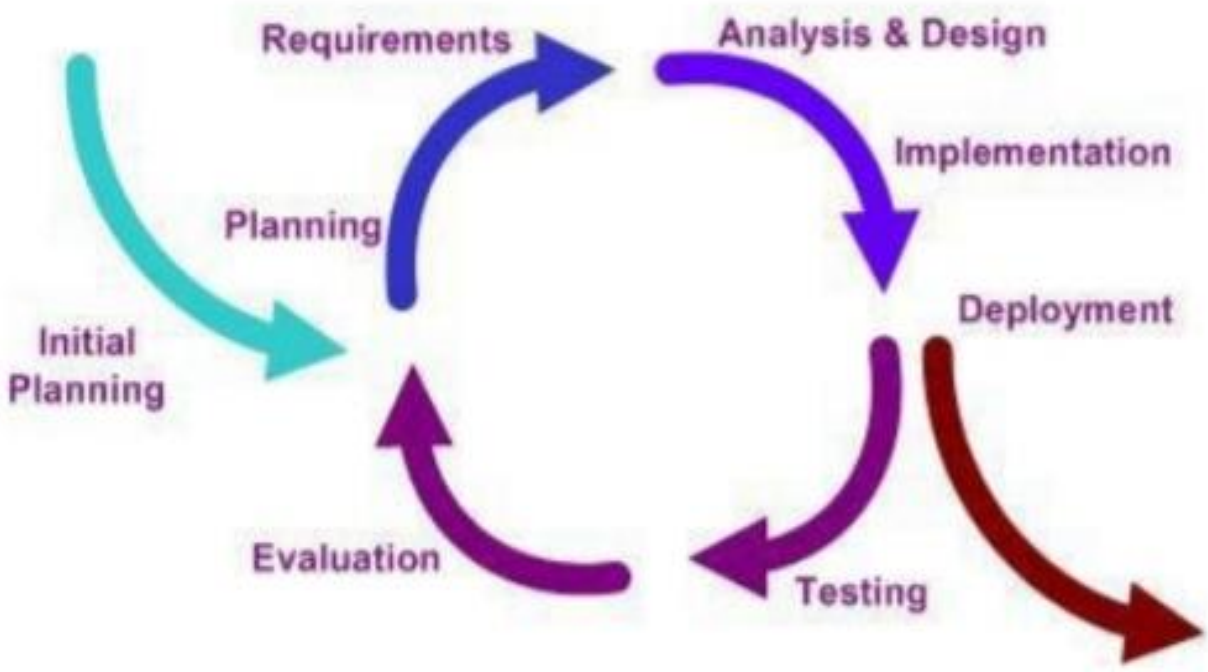
We planned the project over a period of 7 days and divided it into four iterations. We planned the first iteration for analysis, second iteration for game design, third iteration for coding and the final iteration for the product.

In the first iteration, we focused on Project analysis determined as the first planned milestone of the project. Analysis is essential for starting of upcoming milestones and delivering a finished project on time. Successful completion of a project is heavily dependent on effective analysis.

The second iteration started by brainstorming among group members on what the game would be. Each group member denoted the attributes or properties of the game that one dreamed to implement. We gathered suggestions together and chose the ones that was possible to be implemented within a 7-day project time. As soon as the game concept became clear, we made some early decisions on basic requirements of the project to reach the development goals more easily. Game design document was meant to be a living document. In other words, throughout the production process the document was updated, if needed.

In the third iteration coding in C using Unity game engine was under way. Therefore, most of time in this iteration was dedicated for internal training sessions. In this iteration, we needed to achieve four milestones each was dependent on the previous one.

Last iteration was planned for testing and finalizing the product. The testing process is an iterative process. We performed the testing process in four iterations. The successful testing process of software requires a good plan. Therefore, after the requirements of the project are confirmed, the future testing of the system and the code were planned. The test plan provided information on how and when the testing will be executed. In the second iteration, test cases were designed for the planned tests. In iteration three, the designed test cases were executed alongside the module testing and usability testing. During the last iteration, according to the result of the tests, the test reports were documented properly, and the bugs were reported after the testing is completed.



9. Risk Management

A common definition of risk is an uncertain event that if it occurs, can have a positive or negative effect on a project's goals. It helps us to achieve the project's objectives, thus ensuring the successful completion of the project. For successful development of this project, we must need to identify the possible risk. The possible risk for this project is described in below:

(a) The probability of moving away our-self from this project before it is finish is low.
(b) The probability of user acceptance is great.
(c) The probability of requirement cannot come if the time is low.
(d) The probability of marketing the product system is great.
(e) The probability of technology components is not fit for purpose of this project is low.
(f) The probability of selecting low quality requirements is low.
(g) The probability of take wrong decisions are low.
(h) The probability of does not complete this project within a limited time is low.

10. System Specifications

Most of the computer games require high configurations of computer. But in the case of the proposed gaming system, the system requirements are not that much.
The systems requirements for the proposed project "Space Shooter" game are mentioned following.

Operating System:	Windows XP, 7, 8, 8.1, 10
Primary memory (RAM):	1 GB and onwards
Secondary memory (Hard Disk):	100 MB onwards
Processor:	Pentium, Pentium iv,
Core:	i3, core i5, core i7

11. Programming Languages

(a) HTML
(b) JAVASCRIPT
(c)CSS

12. Development Tasks

1. Graphics engine will be responsible for rendering text, 2D images, and 3D models on screen.

(a) Drawing models
(b) Drawing sprites
(c) Drawing text
(d) Texturing models
(e) Animation

2. Sound engine will be responsible for playing music and sound effects.

(a) Multithreading
(b) Playing sounds

3. Input engine will be responsible for transferring mouse and keyboard input upon request to the game engine.

(a) Retrieving Input

4. Menu Engine will handle all menus in game.

13. System Design MVC Model

MVC model is one of the software architectures in the software engineering. Basically, it is consisted of 3 parts.

I. **Model**: package with the application's logic related to data as well as data processing method, which directly manipulate data as well as record the action to be performed with the method of implementation.

II. **View**: the designed display. In general, it does not have any logic programming. But it will refresh the display constantly to show any message or annotation to the user. It also displays the result given by the model through Controller.

III. **Controller**: play a role in the organization between the different levels is used to control the flow of the application. Controller is responsible for controls entire program logic, manages the relationship of objects and handles the event and responds.

14. Testing

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design, and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software. A good test case is one that has a high probability of finding an as undiscovered error.

15. Test Levels

The test approach is divided into three main phases: Module testing, integration testing and system testing. In addition, the system testing includes two sub-phases: functional and usability testing. These planned tests are explained briefly below.

A. **Module testing** will perform during coding by using debug messages to check that the written code produce wanted results. An important requirement is that the code will compile with zero bugs.

B. **Integration testing** will perform after finish module testing to validate if each module can work fine with each other. Integration Test proves that system works as integrated unit when all the fixes are complete.

C. **System testing** includes two phases: functional testing and usability testing. These will perform after the product reaches its final version. During functional test phase, the tester will test if the product meets the game requirements. The tester tests the requirements using the use cases listed below in Test Cases section. The usability test will perform to understand how easy it is to learn to play the game. Any person out of the team members will perform this test by playing the game.

16. Conclusion

(a) The Obstacles

1. Working with game engine completely a new experience for me. Normally I am working with different OO languages, DBMS, mark up languages etc.
2. It is very sensible work, and it demands much time because the game engines try to connect game environment with the real world.
3. Creating a 3d model is very difficult because you need to work with each point of the model.
4. The Exists game engines demands vast knowledge about its properties, sections, and sub-sections.

(b) The Achievements

i. Now I know much more about game engines. How it works? The properties, objects, and others. 2. I know how a model is constructed and how it is animated.
ii. The main thing is that as a software engineer, skill and expertise to create a SRS document and an overall software product report is now better than before.
iii. Develop communication skills
iv. Growing creative thinking and imagination capability.

17. Reference

- You tube
- Play Store
- www.spaceshooter.com/in

18. Bibliography

1. David J Anderson, Kanban: successful evolutionary change for your technology business, Blue Hole Press, 2010.
2. Color Oracle, Design for the Color Impaired, [Online] Available: colororacle.org, last accessed: May 2017.
3. GitHub Inc., GitHub, [Online] Available: www.github.com, last accessed: June 2017

HTML and CSS Code

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Space Shooter</title>
</head>
<style>
body {
width: 100%;
overflow: hidden;
background: url("space1.jpg");
background-size: cover;
}
#board {
position: relative;
width: 600px;
height: 650px;
margin: 50px auto;
border: 3px solid white;
background-color: black;
overflow: hidden;
}
```

```
#jet {
        position: absolute;
        width: 50px;
       height: 50px;
       background: url("rocket.png");
       background-size: contain;
       background-repeat: no-repeat;
       left: 50%;
        bottom: 0px;
}

.bullets {
              position: absolute;
              bottom: 40px;
                left: 50%;
                width: 20px;
               height: 20px;
               background: url("bullet.png");
                background-size: contain;
               background-repeat: no-repeat;
          }
```

```
.rocks {
        position: absolute;
        top: 0px;
        width: 60px;
        height: 60px;
        background: url("alien.png");
        background-size: contain;
        background-repeat: no-repeat;
    }

#points {
        position: absolute;
        top: 0px;
        margin: 40px;
        opacity: 0.9;
        font-size: 10em;
        color: white;
                }
  </style>
  <script src="app.js" defer></script>
  <body>

<audio autoplay>
<source src="Star Wars.mp3" type="audio/mp3">
</audio>
```

```
<div id="board">
     <div class="rocks"></div>
     <div id="jet"></div>
   </div>
   <div id="points">0</div>

 </body>
</html>
```

JAVA Script Code

```
var jet = document.getElementById("jet");
var board = document.getElementById("board");


window.addEventListener("keydown", (e) => {
 var left =
parseInt(window.getComputedStyle(jet).getPropertyValue("left"));
 if (e.key == "ArrowLeft" && left > 0) {
  jet.style.left = left - 10 + "px";
 }
 //569  =>  board width - jet width
 else if (e.key == "ArrowRight" && left <= 569) {
  jet.style.left = left + 10 + "px";
 }


 if (e.key == "ArrowUp" || e.keyCode == 32) {
  //32 is for space key
  var bullet = document.createElement("div");
  bullet.classList.add("bullets");
  board.appendChild(bullet);
```

```
  var movebullet = setInterval(() => {
    var rocks = document.getElementsByClassName("rocks");

    for (var i = 0; i < rocks.length; i++) {
      var rock = rocks[i];
      if (rock != undefined) {
        var rockbound = rock.getBoundingClientRect();
        var bulletbound = bullet.getBoundingClientRect();

        //Condition to check whether the rock/alien and the bullet are at the same position..!
        //If so,then we have to destroy that rock

        if (
          bulletbound.left >= rockbound.left &&
          bulletbound.right <= rockbound.right &&
          bulletbound.top <= rockbound.top &&
          bulletbound.bottom <= rockbound.bottom
        ) {
          rock.parentElement.removeChild(rock); //Just removing that particular rock;
          //Scoreboard
          document.getElementById("points").innerHTML =
            parseInt(document.getElementById("points").innerHTML) + 1;
```

```
                    }
                  }
                }
                    var bulletbottom = parseInt(
      window.getComputedStyle(bullet).getPropertyValue("bottom")
    );

    //Stops the bullet from moving outside the gamebox
    if (bulletbottom >= 650) {
      clearInterval(movebullet);
    }

    bullet.style.left = left + "px"; //bullet should always be placed at the top of my jet..!
    bullet.style.bottom = bulletbottom + 3 + "px";
  });
 }
});

var generaterocks = setInterval(() => {
 var rock = document.createElement("div");
 rock.classList.add("rocks");
 //Just getting the left of the rock to place it in random position...
 var rockleft = parseInt(
```

```
 window.getComputedStyle(rock).getPropertyValue("left")
  );
  //generate value between 0 to 450 where 450 => board width - rock width
  rock.style.left = Math.floor(Math.random() * 530) + "px";

  board.appendChild(rock);
   }, 1000);

var moverocks = setInterval(() => {
  var rocks = document.getElementsByClassName("rocks");

  if (rocks != undefined) {
    for (var i = 0; i < rocks.length; i++) {
      //Now I have to increase the top of each rock,so that the rocks can move
downwards..
      var rock = rocks[i]; //getting each rock
      var rocktop = parseInt(
        window.getComputedStyle(rock).getPropertyValue("top")
      );
```

```
//620 => boardheight - rockheight + 25
if (rocktop >= 620) {
  alert("Game Over");
  clearInterval(moverocks);
  window.location.reload();
}

rock.style.top = rocktop + 26 + "px";
                  }
                 }
               }, 450);
```

Thank You!