```java
/*
    MJKP 2020 ROCKET LAUNCHING
    --------------------------
    TEAM MEMBERS
    --------------------------
    Jomi Susan Mathew       36
    M Aswin Kishore         41
    Karthika Sankar         39
    Parvathy S Thampi       49
    --------------------------

    Build on JDK 8.0
    Java version: 1.8.0_271

    --------------------------

*/

import java.awt.*;
import java.awt.event.*;
import java.util.Random;
import javax.swing.*;

class Main extends JFrame implements ActionListener
{
    // Coordinates of rocket
    static int xRocket = 300;
    static int yRocket = 300;

    // To determine the number of stars
    int timer = 5;

    // Importing images
    Image rocket = Toolkit.getDefaultToolkit().getImage("rocket.png");
    Image rocket_NoEngine = Toolkit.getDefaultToolkit().getImage("rocket_noengine.png");

    Button liftOff;
    Label Data;

    boolean Takeoff = false;

    // Threads to repaint the screen and to Increment Timers
    RepaintThread launch = new RepaintThread();
    TimerThread LoopTimer = new TimerThread();

    public Main()
    {
        Data = new Label("MJKP 2020");
        Data.setBounds(840, 100, 110, 40);
        Data.setFont(new Font("Times New Roman", Font.BOLD, 20));
        Data.setForeground(Color.WHITE);
        Data.setBackground(Color.BLACK);
        add(Data);

        liftOff = new Button("START/STOP");
        liftOff.setBounds(820, 200, 150, 40);
        liftOff.setFont(new Font("Times New Roman", Font.BOLD, 20));
        liftOff.setForeground(Color.WHITE);
        liftOff.setBackground(Color.BLACK);
        liftOff.addActionListener(this);
        add(liftOff);

        Background panel = new Background();
        add(panel);

        setBackground(Color.BLACK);
        setTitle("MJKP 2020 Rocket Launching");
        setResizable(false);
        setSize(1000, 800);
        setVisible(true);

    }

    public void actionPerformed(ActionEvent e) // Called by pressing the button START/STOP
```

```
    {
        if (Takeoff == false)
        {
            Takeoff = true;
            liftOff.setForeground(Color.ORANGE); // Set color of button to Orange

            try
            {
                launch.start();
                LoopTimer.start();
            }
            catch (Exception d)
            {
            }
        }
        else
        {
            // Setting Default Values
            Takeoff = false;
            liftOff.setForeground(Color.WHITE); // Set color of button to White
            timer = 10;
            xRocket = 300;
            yRocket = 300;

            try
            {
                launch.interrupt();
                LoopTimer.interrupt();
            }
            catch (Exception d)
            {
            }
        }

    }

    // Thread to repaint the Frame
    class RepaintThread extends Thread
    {
        public void run()
        {
            while (true)
            {
                try
                {
                    repaint();
                    Thread.sleep(17);
                }
                catch (Exception e)
                {
                }
            }
        }
    }

    // Thread to increment timer in Loop
    class TimerThread extends Thread
    {
        public void run()
        {
            while (true)
            {
                if (timer >= 300)
                    timer = 300;
                else
                    timer += 1;

                try
                {
                    Thread.sleep(400);
                }
                catch (Exception e)
                {
                }

```

```
            }
        }
    }

    class Background extends JPanel implements MouseMotionListener
    {

        public Background()
        {
            addMouseMotionListener(this);
        }

        @Override
        public void paintComponent(Graphics g)
        {
            Graphics sky = getGraphics();

            if (!Takeoff) // Before TakeOff
            {
                g.setColor(Color.BLACK);
                g.fillRect(0, 0, 800, 800);
                g.drawImage(rocket_NoEngine, 300, 300, this);
            }
            else // After TakeOff
            {
                g.setColor(Color.BLACK);
                g.fillRect(0, 0, 800, 800);
                g.drawImage(rocket, xRocket, yRocket, this);

                for (int i = 0; i < timer; i++) // Generating Stars Loop
                {
                    Random num = new Random(); // To create random X and Y coordinates to set the stars location

                    sky.setColor(Color.WHITE);
                    sky.fillOval(num.nextInt(800), num.nextInt(800), 3, 3);
                }

                try
                {
                    Thread.sleep(17);
                }
                catch (Exception e)
                {
                }

            }
        }

        public void mouseDragged(MouseEvent e)
        {
        }

        public void mouseMoved(MouseEvent e)
        {
            if (Takeoff)
            {
                // Limit is set so that mouse location does not go beyond the Rectangular box
                if (e.getX() <= 670 && e.getY() <= 700)
                {
                    xRocket = e.getX() - 100;
                    yRocket = e.getY() - 5;
                }
            }
        }
    }

    public static void main(String args[])
    {
        new Main();
    }
}
```