



Student ID: 0806797

Student Name: Aidan Morbi

Program Testing: Screenshot-based functionality testing with a short description of each test.
Remember to test every edge case (e.g., CSV does not exist):

Copy and paste your code below:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class App extends JFrame {
    // Declaring a JTextArea
    private JTextArea textArea;

    // App Class Constructor.
    public App() {
        // JFrame settings
        // Setting Title
        setTitle("Aidan's CSV Loader Application");
        // Setting Dimensions of Window
        setSize(500, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        /*
         * 1-Create a new button named LoadButton. The text in the Button should
say
         * LoadCSV
         * 2-After that initialize a new TextArea (we already declared a textArea
on
         * Line 11), this is the area where we will display our CSV.
        */
    }
}
```

```

        * 3-Set the new textArea to be uneditable (e.g., .setEditable(false))
        * 4-Create a JScrollPane within the text area, so that we can scroll up
and
        * down.
        * 5-Add an action listener to the button that calls the Load CSV
function
        * (below)
        * 6-Play around with coloring and styling to make your application look
more
        * professional
        */

        JButton loadButton = new JButton("Load CSV");
        textArea = new JTextArea();
        textArea.setEditable(false); // uneditable
        JScrollPane scrollPane = new JScrollPane(textArea); // Wrapping text area
in a JScrollPane

        // styling
        textArea.setBackground(Color.LIGHT_GRAY);
        loadButton.setForeground(Color.BLUE);
        loadButton.setBackground(Color.WHITE);

        // Action Listener to Load button
        loadButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFileChooser fileChooser = new JFileChooser();
                int returnValue = fileChooser.showOpenDialog(null);
                if (returnValue == JFileChooser.APPROVE_OPTION) {
                    String filePath =
fileChooser.getSelectedFile().getAbsolutePath();
                    loadCsv(filePath);
                }
            }
        });

        add(loadButton, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);
    }

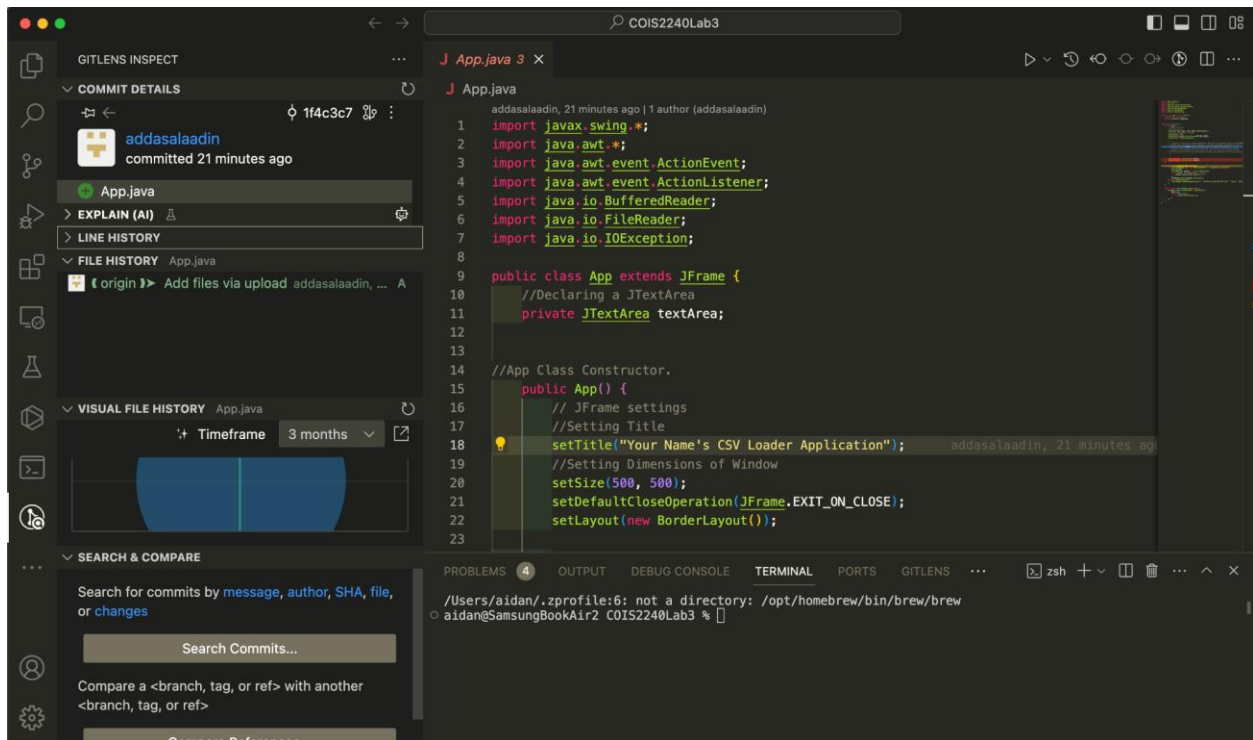
    private void loadCsv(String filePath) {
        try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
            String line;
            StringBuilder content = new StringBuilder();

```

```
        while ((line = br.readLine()) != null) {
            content.append(line).append("\n");
        }
        textArea.setText(content.toString());
    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "Failed to load the CSV file.",
"Error", JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new App().setVisible(true);
        }
    });
}
}
```

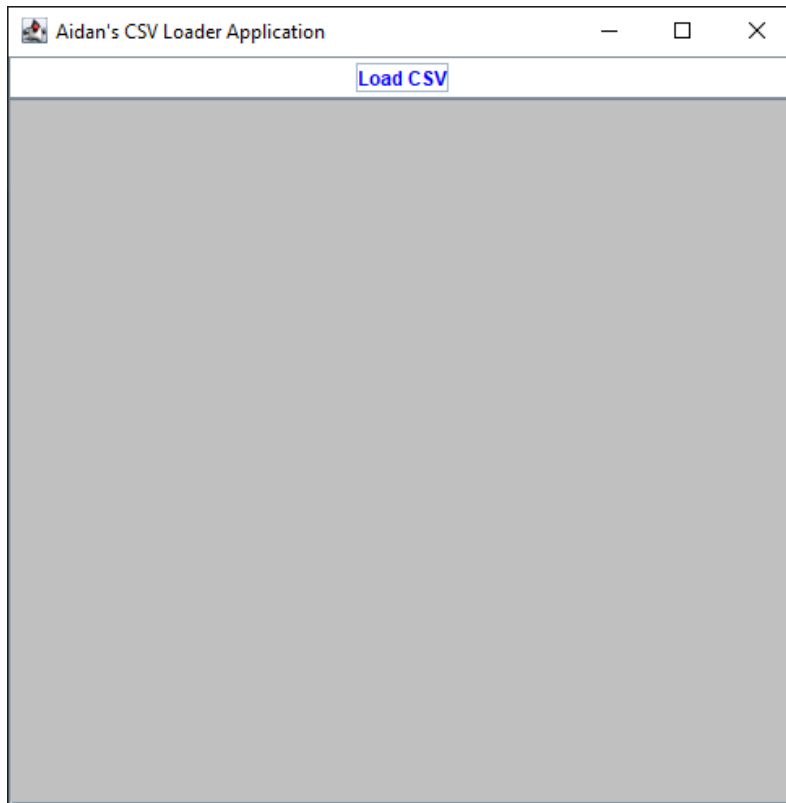
Prove using a screenshot that you cloned the GitHub repository:



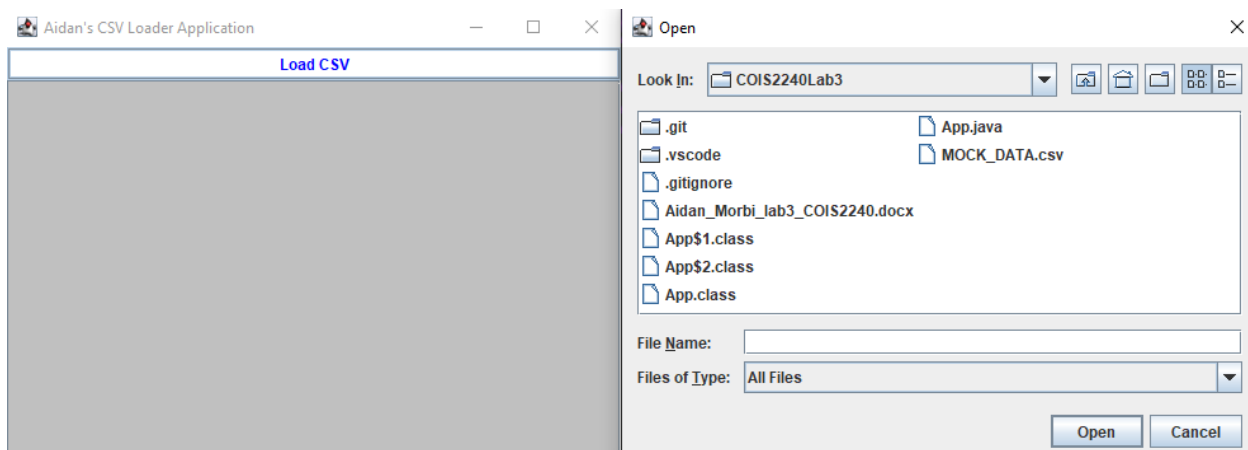
Testing:

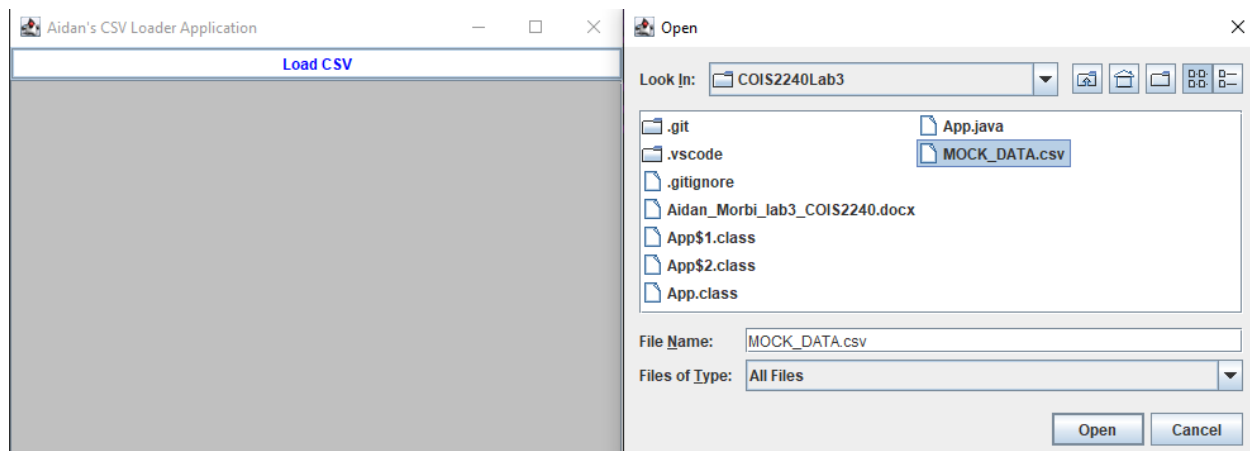
Test 1, proper usage:

On run:

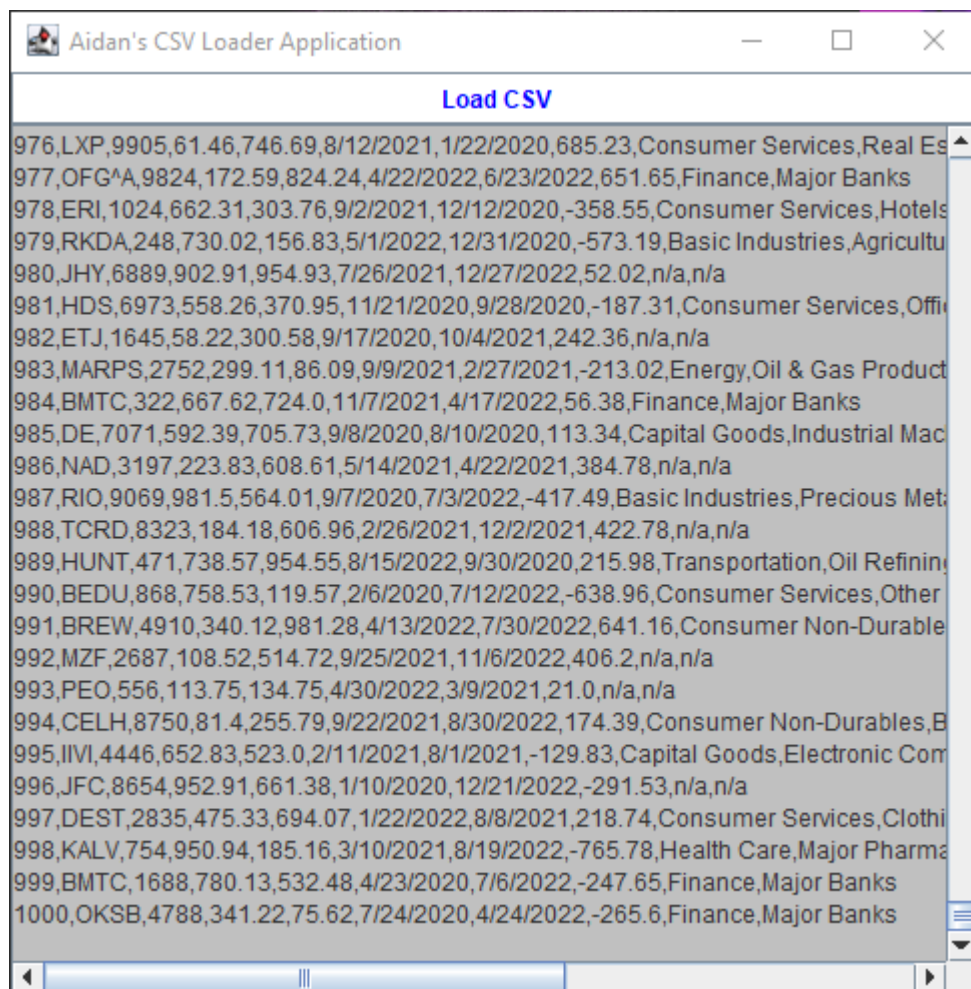


On load CSV:



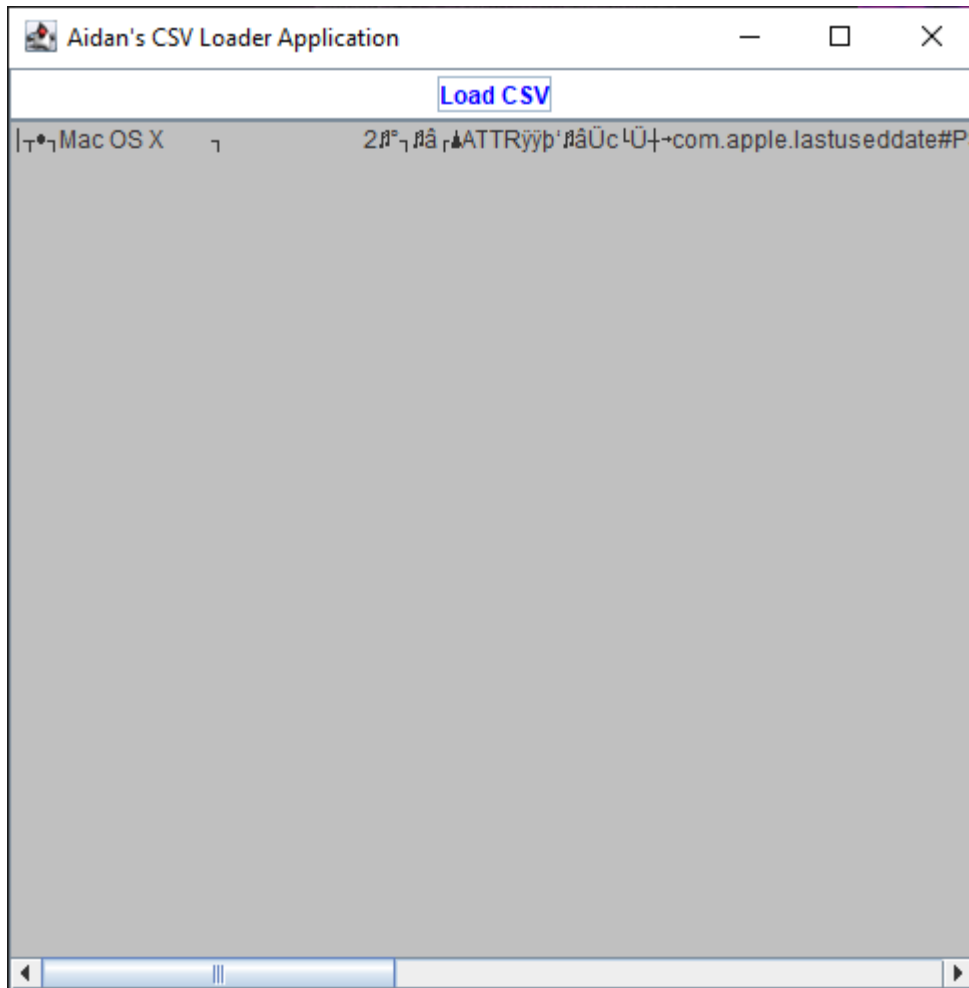


Open Clicked:

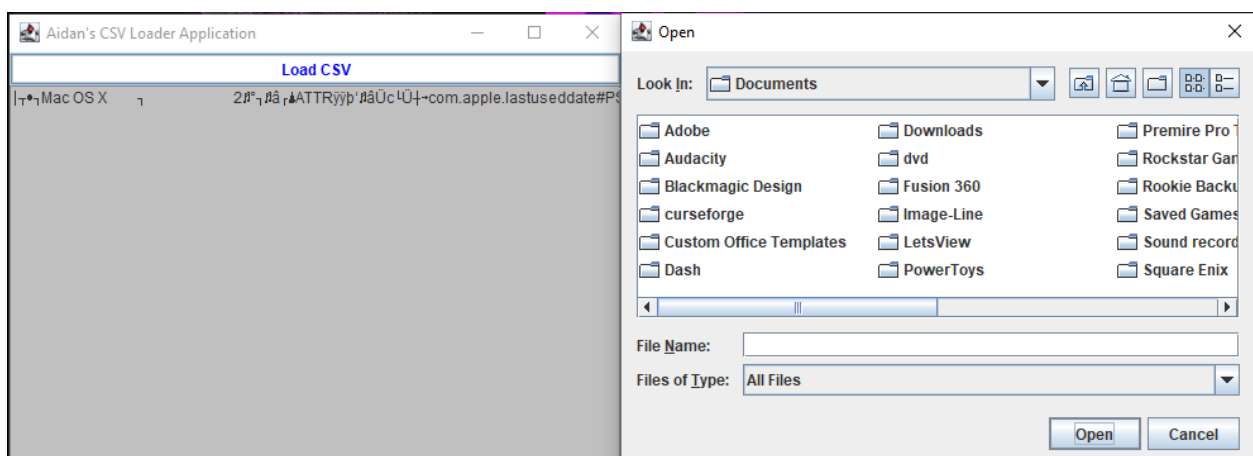


Test 2, Incorrect file type:

*Tried to load video file:



Test 3, Opening nothing:



The program does not let the user do this.

Further potential issues include but are not necessarily limited to:

- Issues with file encoding due to char set
- CSV format variation (here it is loaded as plain text)
- Memory usage due to the entire content of the CSV file being loaded in to memory within the StringBuilder and then set in the JTextArea. This could cause high memory usage
- Large CSV files
 - This is mainly an issue with extremely large files however as we know many companies things that excel is a database so it wouldn't be surprising to have a multibillion row csv file which would cause issues