EQUIPMENT RENTAL SYSTEM

Submitted by

KARTHIKEYA DEVARLA [RA2211003010192]
SIDDARTH JAVVADI [RA2211003010154]
GOWTHAM REDDY KOMMEPALLI [RA2211003010169]

Under the Guidance of

Mr. ARULALAN V

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELORS OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

NOVEMBER 2023

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled **"EQUIPMENT RENTAL SYSTEM"** is the bonafide work done by **KARTHIKEYA DEVARLA [RA2211003010192], SIDDARTH JAVVADI [RA2211003010154] and GOWTHAM REDDY KOMMEPALLI [RA2211003010169]** who carried out under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Faculty In-Charge
Mr. Arulalan V
Assistant Professor
Department of Computing Technologies
SRM Institute of Science and Technology

HEAD OF THE DEPARTMENT
Dr. M. Pushpalatha
Professor and Head
Department of Computing Technologies
SRM Institute of Science and Technology

TABLE OF CONTENTS

1. INTRODUCTION

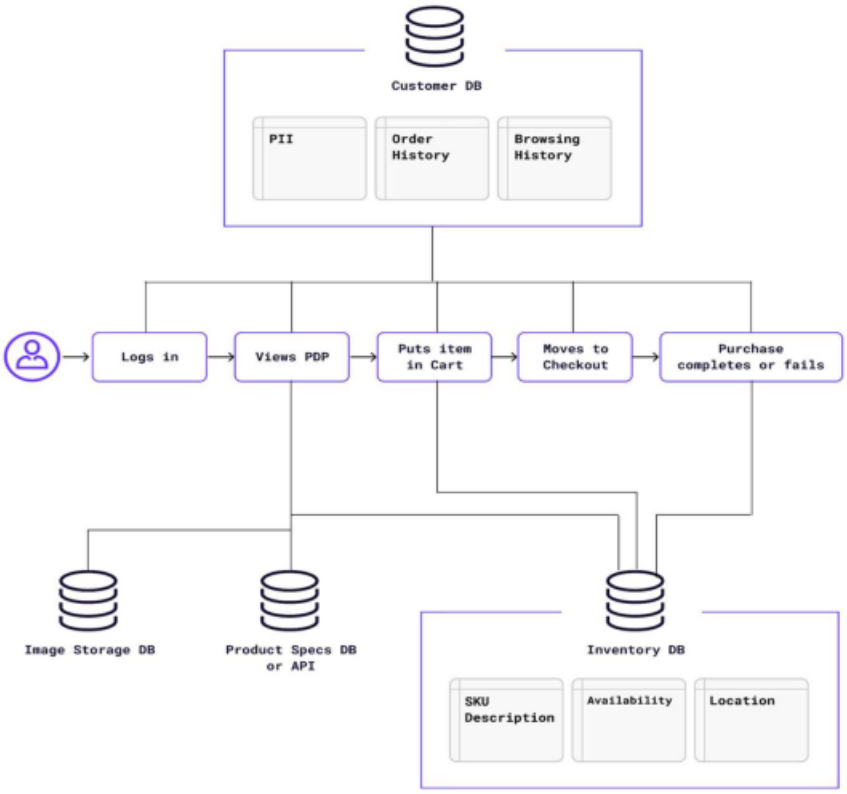
In an era where traditional ownership models are giving way to shared economy practices, our Equipment Lending and Rental System represents a visionary leap towards a more sustainable, cost-effective, and interconnected future. Today, our rapidly evolving world demands flexible and innovative solutions that cater to the ever-changing needs of students and faculty. The old adage of "use it or lose it" no longer applies in a society increasingly conscious of resource utilization and economic efficiency. With our platform, we empower users with the ability to access the tools and equipment they require precisely when they need them, freeing them from the financial burden and environmental impact of long-term ownership. This shift towards access over ownership is not just a pragmatic choice; it's a statement of responsibility towards our shared environment.

Furthermore, our Equipment Lending and Rental System goes beyond the simple exchange of physical items. It seeks to redefine the very essence of the university experience by facilitating not only the sharing of equipment but also the exchange of knowledge and the cultivation of a sense of unity and collective responsibility. In the collaborative world we envision, students, faculty, and local businesses become integral parts of a vibrant ecosystem, where the flow of resources and information strengthens the bonds of our academic communities. With seamless digital interfaces and secured transactions, we provide the infrastructure that underpins a transformational approach to university life. Our vision is one where university communities not only thrive through collaborative consumption but also actively contribute to both individual growth and the betterment of our environment, forging a path to a brighter and more sustainable future.

2. ARCHITECTURE AND DESIGN

The design of our project revolves around creating an advanced equipment rental system tailored for universities. With a user-centric approach, our system offers a wide-ranging equipment catalog and streamlined reservation processes. Real-time inventory management and notifications ensure equipment availability and timely returns. Integration with university systems, mobile accessibility, and multilingual support enhance user convenience. Through a combination of innovation and user-friendly design, we aim to provide a comprehensive and efficient solution for universities' equipment rental needs

ARCHITECTURE DIAGRAM



Architecture Diagram

CODE

```
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;

public class EquipLendGUI {
    private final JFrame frame;
    private JTextField emailTextField;
    private JPasswordField passwordTextField;
    private JTextField nameTextField;
    private JTextField mobileNumberTextField;
    private JPasswordField confirmPasswordTextField;

    public EquipLendGUI() {
        frame = new JFrame("Equipment Lending and Rental System");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 300);
        frame.setLayout(new BorderLayout());

        JPanel panel = new JPanel(new GridBagLayout());
        frame.add(panel, BorderLayout.CENTER);

        GridBagConstraints constraints = new GridBagConstraints();
        constraints.fill = GridBagConstraints.HORIZONTAL;
        constraints.insets = new Insets(10, 10, 10, 10);

        JLabel descriptionLabel = new JLabel("<html>Welcome to the Equipment Lending and
Rental System!<br/>Please select an option below:</html>");
        descriptionLabel.setFont(new Font("Arial", Font.PLAIN, 16));
        descriptionLabel.setHorizontalAlignment(JLabel.CENTER);
```

```
    constraints.gridx = 0;
    constraints.gridy = 0;
    constraints.gridwidth = 2;
    panel.add(descriptionLabel, constraints);

    emailTextField = new JTextField(20);
    passwordTextField = new JPasswordField(20);
    nameTextField = new JTextField(20);
    mobileNumberTextField = new JTextField(20);
    confirmPasswordTextField = new JPasswordField(20);

    JButton loginButton = createButton("Login");
    constraints.gridy = 1;
    panel.add(loginButton, constraints);

    JButton signupButton = createButton("Sign Up");
    constraints.gridy = 2;
    panel.add(signupButton, constraints);

    frame.setVisible(true);
}

private JButton createButton(String label) {
    JButton button = new JButton(label);
    button.setFont(new Font("Arial", Font.PLAIN, 16));
    button.setPreferredSize(new Dimension(120, 40));
    button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (label.equals("Login")) {
                openLoginFrame();
            } else if (label.equals("Sign Up")) {
                openSignupFrame();
            }
        }
```

```
        });
        return button;
    }
    /*   public addLabelAndField(JPanel panel, GridBagConstraints constraints, String label,
int row, JTextField textField) {
        JLabel fieldLabel = new JLabel(label);
        constraints.gridx = 0;
        constraints.gridy = row;
        panel.add(fieldLabel, constraints);

        constraints.gridx = 1;
        constraints.gridy = row;
        panel.add(textField, constraints);
    }
*/
    private void openLoginFrame() {
        JFrame loginFrame = new JFrame("Login Page");
        loginFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel loginPanel = new JPanel(new GridBagLayout());
        loginFrame.add(loginPanel);

        GridBagConstraints constraints = new GridBagConstraints();
        constraints.fill = GridBagConstraints.HORIZONTAL;
        constraints.insets = new Insets(10, 10, 10, 10);

        addLabelAndField(loginPanel, constraints, "Email:", 0, emailTextField);
        addLabelAndField(loginPanel, constraints, "Password:", 1, passwordTextField);

        JButton submitButton = new JButton("Submit");
        constraints.gridx = 0;
        constraints.gridy = 2;
        constraints.gridwidth = 2;
        constraints.anchor = GridBagConstraints.CENTER;
        loginPanel.add(submitButton, constraints);
```

```
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String email = emailTextField.getText();
                String password = new String(passwordTextField.getPassword());

                // Add logic to check login credentials here
                // For example:
                // if (checkLoginCredentials(email, password)) {
                //     JOptionPane.showMessageDialog(loginFrame, "Login successful!");
                //     openWelcomeFrame("User");
                //     loginFrame.dispose();
                // } else {
                //     JOptionPane.showMessageDialog(loginFrame, "Login failed. Please check
your credentials.");
                // }
            }
        });
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // You can bypass the database check and directly open the welcome frame.
                openWelcomeFrame("User");
                loginFrame.dispose();
            }
        });


        loginFrame.pack();
        loginFrame.setLocationRelativeTo(null);
        loginFrame.setVisible(true);
    }

    /*  private void addLabelAndField(JPanel panel, GridBagConstraints constraints, String
```

```
label, int row, JTextField textField) {
        JLabel fieldLabel = new JLabel(label);
        constraints.gridx = 0;
        constraints.gridy = row;
        panel.add(fieldLabel, constraints);

        constraints.gridx = 1;
        constraints.gridy = row;
        panel.add(textField, constraints);
    }*/

    private void openSignupFrame() {
        JFrame signupFrame = new JFrame("Sign Up Page");
        signupFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel signupPanel = new JPanel(new GridBagLayout());
        signupFrame.add(signupPanel);

        GridBagConstraints constraints = new GridBagConstraints();
        constraints.fill = GridBagConstraints.HORIZONTAL;
        constraints.insets = new Insets(10, 10, 10, 10);

        addLabelAndField(signupPanel, constraints, "Name:", 0, nameTextField);
        addLabelAndField(signupPanel, constraints, "Email:", 1, emailTextField);
        addLabelAndField(signupPanel, constraints, "Mobile Number:", 2,
mobileNumberTextField);

        JLabel roleLabel = new JLabel("Role:");
        String[] roles = {"Student", "Faculty", "Vendor"};
        JComboBox<String> roleComboBox = new JComboBox<>(roles);
        constraints.gridx = 0;
        constraints.gridy = 3;
        signupPanel.add(roleLabel, constraints);
        constraints.gridx = 1;
        signupPanel.add(roleComboBox, constraints);
```

```
        addLabelAndField(signupPanel, constraints, "Password:", 4, passwordTextField);
        addLabelAndField(signupPanel, constraints, "Re-enter Password:", 5,
confirmPasswordTextField);

        JButton submitButton = new JButton("Submit");
        constraints.gridx = 0;
        constraints.gridy = 6;
        constraints.gridwidth = 2;
        constraints.anchor = GridBagConstraints.CENTER;
        signupPanel.add(submitButton, constraints);

        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String name = nameTextField.getText();
                String email = emailTextField.getText();
                String mobileNumber = mobileNumberTextField.getText();
                String role = roleComboBox.getSelectedItem().toString();
                String password = new String(passwordTextField.getPassword());
                String confirmPassword = new String(confirmPasswordTextField.getPassword());

                // Add logic to register the user here
                // For example:
                // if (registerUser(name, email, mobileNumber, role, password, confirmPassword))
{
                //     JOptionPane.showMessageDialog(signupFrame, "Registration successful!");
                //     openWelcomeFrame(name);
                //     signupFrame.dispose();
                // } else {
                //     JOptionPane.showMessageDialog(signupFrame, "Registration failed. Please
check your input.");
                // }
            }
        });
```

```
    submitButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Bypass user registration logic and directly open the welcome frame.
            openWelcomeFrame(nameTextField.getText());
            signupFrame.dispose();
        }
    });


    signupFrame.pack();
    signupFrame.setLocationRelativeTo(null);
    signupFrame.setVisible(true);
}

/**
 * @param userName
 */
private void openWelcomeFrame(String userName) {
    JFrame welcomeFrame = new JFrame("Welcome");
    welcomeFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    JPanel welcomePanel = new JPanel(new GridBagLayout());
    welcomeFrame.add(welcomePanel);

    GridBagConstraints constraints = new GridBagConstraints();
    constraints.fill = GridBagConstraints.HORIZONTAL;
    constraints.insets = new Insets(10, 10, 10, 10);

    JLabel greetingLabel = new JLabel("Welcome, " + userName + "!");
    greetingLabel.setFont(new Font("Arial", Font.BOLD, 16));
    greetingLabel.setHorizontalAlignment(JLabel.CENTER);
    constraints.gridx = 0;
    constraints.gridy = 0;
```

```
constraints.gridwidth = 2;
constraints.anchor = GridBagConstraints.CENTER;
welcomePanel.add(greetingLabel, constraints);

JLabel questionLabel = new JLabel("Are you looking to lend or borrow?");
questionLabel.setFont(new Font("Arial", Font.PLAIN, 16));

JButton lendButton = new JButton("Lend");
JButton borrowButton = new JButton("Borrow");

constraints.gridx = 0;
constraints.gridy = 1;
constraints.gridwidth = 2;
constraints.anchor = GridBagConstraints.CENTER;
welcomePanel.add(questionLabel, constraints);

constraints.gridx = 0;
constraints.gridy = 2;
constraints.gridwidth = 1;
constraints.anchor = GridBagConstraints.EAST;
welcomePanel.add(lendButton, constraints);

constraints.gridx = 1;
constraints.gridy = 2;
constraints.gridwidth = 1;
constraints.anchor = GridBagConstraints.WEST;
welcomePanel.add(borrowButton, constraints);

lendButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        openLendItemsPage();
        welcomeFrame.dispose();
    }
});
```

```
    borrowButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            openBorrowFrame();
            welcomeFrame.dispose();
        }
    });

    welcomeFrame.pack();
    welcomeFrame.setLocationRelativeTo(null);
    welcomeFrame.setVisible(true);
}

private void openLendItemsPage() {
    JFrame lendingFrame = new JFrame("Lend a Product");
    lendingFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    JPanel lendingPanel = new JPanel(new GridBagLayout());
    lendingFrame.add(lendingPanel);

    GridBagConstraints constraints = new GridBagConstraints();
    constraints.fill = GridBagConstraints.HORIZONTAL;
    constraints.insets = new Insets(10, 10, 10, 10);

    JLabel productNameLabel = new JLabel("Product Name:");
    JTextField productNameField = new JTextField(20);
    addLabelAndField(lendingPanel, constraints, "Product Name:", 0, productNameField);

    JLabel descriptionLabel = new JLabel("Description:");
    JTextField descriptionField = new JTextField(20);
    addLabelAndField(lendingPanel, constraints, "Description:", 1, descriptionField);

    JLabel conditionLabel = new JLabel("Condition:");
    JTextField conditionField = new JTextField(20);
    addLabelAndField(lendingPanel, constraints, "Condition:", 2, conditionField);
```

```
        JLabel costLabel = new JLabel("Cost per day:");
        JTextField costField = new JTextField(20);
        addLabelAndField(lendingPanel, constraints, "Cost per day:", 3, costField);

        JLabel categoryLabel = new JLabel("Category:");
        String[] categories = {"Electronics", "Mobility", "Books", "Hardware Tools",
"Stationary", "Sports Gear"};
        JComboBox<String> categoryComboBox = new JComboBox<>(categories);
        constraints.gridx = 0;
        constraints.gridy = 4;
        lendingPanel.add(categoryLabel, constraints);
        constraints.gridx = 1;
        lendingPanel.add(categoryComboBox, constraints);

        JButton submitButton = new JButton("Submit");
        constraints.gridx = 0;
        constraints.gridy = 5;
        constraints.gridwidth = 2;
        constraints.anchor = GridBagConstraints.CENTER;
        lendingPanel.add(submitButton, constraints);

        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String productName = productNameField.getText();
                String description = descriptionField.getText();
                String condition = conditionField.getText();
                String costPerDay = costField.getText();
                String selectedCategory = categoryComboBox.getSelectedItem().toString();

                // Add your logic to save the product to your database or perform other actions here
                // For this example, we will just display the entered details
                String message = "Product Name: " + productName + "\nDescription: " +
description + "\nCondition: " + condition
```

```
                + "\nCost per day: " + costPerDay + "\nCategory: " + selectedCategory;
            JOptionPane.showMessageDialog(lendingFrame, message);

            // Optionally, you can clear the fields after submission
            productNameField.setText("");
            descriptionField.setText("");
            conditionField.setText("");
            costField.setText("");
        }
    });

    lendingFrame.pack();
    lendingFrame.setLocationRelativeTo(null);
    lendingFrame.setVisible(true);
}

private void addLabelAndField(JPanel panel, GridBagConstraints constraints, String label,
int row, JTextField textField) {
    JLabel fieldLabel = new JLabel(label);
    constraints.gridx = 0;
    constraints.gridy = row;
    panel.add(fieldLabel, constraints);

    constraints.gridx = 1;
    constraints.gridy = row;
    panel.add(textField, constraints);
}

private void openBorrowFrame() {
    JFrame borrowFrame = new JFrame("Borrow Items");
    borrowFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    JPanel borrowPanel = new JPanel(new BorderLayout());
    borrowFrame.add(borrowPanel);
```

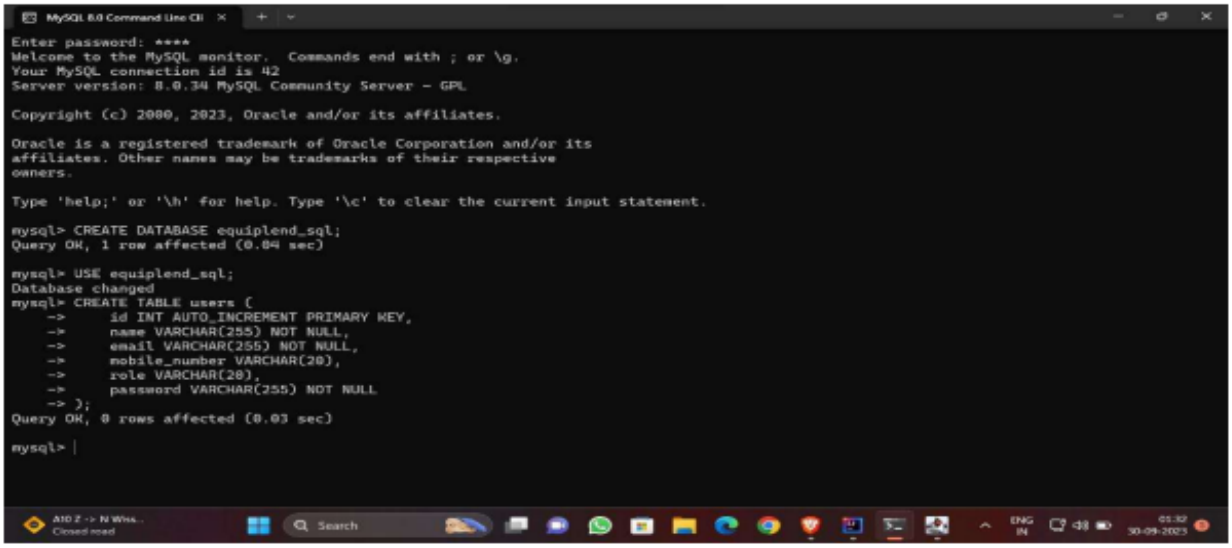
```
        JTextArea borrowDescription = new JTextArea();
        borrowDescription.setText("Welcome to the Borrow Items page. Use the search box
below to find items you want to borrow.");
        borrowDescription.setFont(new Font("Arial", Font.PLAIN, 16));
        borrowDescription.setWrapStyleWord(true);
        borrowDescription.setLineWrap(true);
        borrowDescription.setOpaque(false);
        borrowDescription.setEditable(false);
        borrowPanel.add(borrowDescription, BorderLayout.NORTH);

        JPanel searchPanel = new JPanel();
        searchPanel.setLayout(new BoxLayout(searchPanel, BoxLayout.X_AXIS));
        borrowPanel.add(searchPanel, BorderLayout.CENTER);

        Border searchBorder = BorderFactory.createTitledBorder("Search for Items");
        searchPanel.setBorder(searchBorder);

        JTextField searchField = new JTextField(20);
        searchField.setMaximumSize(searchField.getPreferredSize());
        JButton searchButton = new JButton("Search");

        searchPanel.add(searchField);
        searchPanel.add(searchButton);
// From here
        searchButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String searchQuery = searchField.getText();

                // Implement your search logic here, and display the results to the user
                // For this example, we will just display the search query
                JOptionPane.showMessageDialog(borrowFrame, "Searching for: " + searchQuery);

                // Simulate search results (replace with your actual search logic)
                List<String> searchResults = new ArrayList<>();
```

```
            searchResults.add("Result 1");
            searchResults.add("Result 2");

            // Call the method to display the search results
            displaySearchResults(searchResults);
        }
    });

//Till here
    borrowFrame.pack();
    borrowFrame.setLocationRelativeTo(null);
    borrowFrame.setVisible(true);
}
private void displaySearchResults(List<String> searchResults) {
    JFrame resultFrame = new JFrame("Search Results");
    resultFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    // Create a panel to display search results
    JPanel resultPanel = new JPanel();
    resultPanel.setLayout(new BoxLayout(resultPanel, BoxLayout.Y_AXIS));

    JLabel countLabel = new JLabel("The following items are found:");
    countLabel.setFont(new Font("Arial", Font.BOLD, 16));
    resultPanel.add(countLabel);

    // Add search results to the panel
    for (String result : searchResults) {
        JLabel resultLabel = new JLabel(result);
        resultLabel.setFont(new Font("Arial", Font.PLAIN, 16));
        resultPanel.add(resultLabel);
    }

    JScrollPane resultScrollPane = new JScrollPane(resultPanel);
    resultFrame.add(resultScrollPane);
```

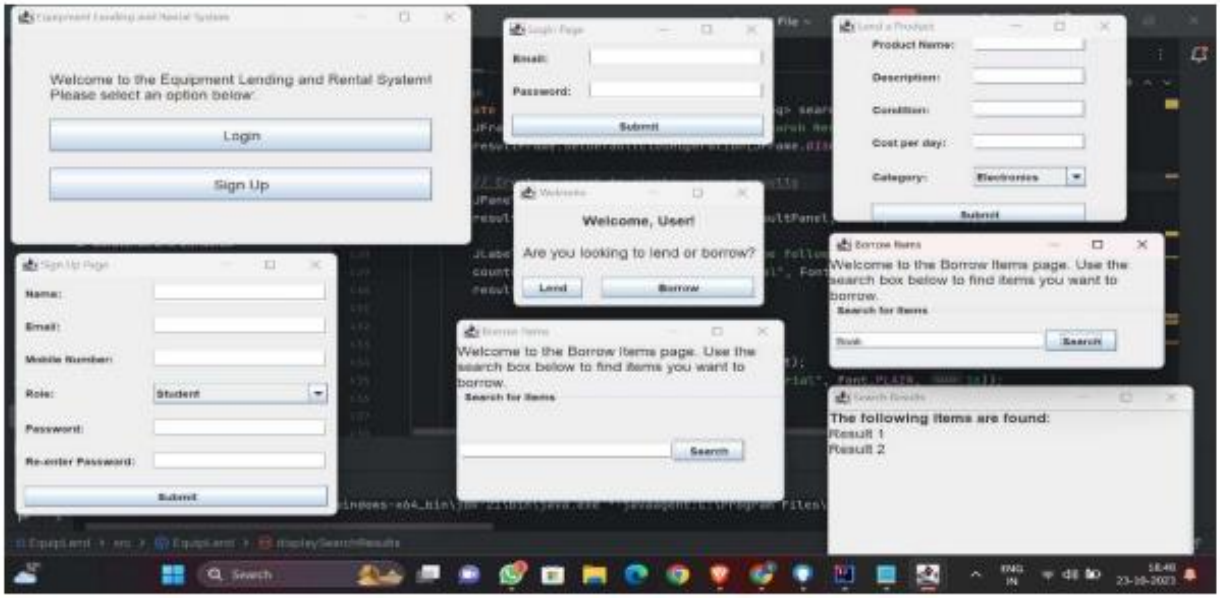
```
        resultFrame.setSize(400, 300); // Adjust the size as needed
        resultFrame.setLocationRelativeTo(null);
        resultFrame.setVisible(true);
    }


    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new EquipLendGUI();
            }
        });
    }
}
```

4. RESULTS AND DISCUSSION

JAVA



SQL Data Base Visualization



5. CONCLUSION

The Equipment Lending and Rental System represents a profound step forward in reshaping our relationship with resources and our environment. As we've explored throughout this report, this innovative platform offers a vision of a more sustainable, cost-effective, and interconnected future. It redefines ownership and access, empowering university communities to make efficient use of equipment, fostering knowledge exchange, and cultivating a sense of unity. This concluding section summarizes the key takeaways and the broader implications of this groundbreaking initiative.

Our project encourages the shift from traditional ownership models to shared economy practices, aligning with the global call for sustainability. By reducing overconsumption and minimizing waste, we actively contribute to a healthier environment. The potential reduction in the production of underutilized items and the ensuing decline in their disposal can have far-reaching environmental benefits.

One of the most tangible advantages of the Equipment Lending and Rental System is the financial relief it offers to students and faculty. The burden of purchasing and maintaining costly equipment is alleviated, enabling them to make more informed financial decisions. By accessing the tools they need without the long-term commitment of ownership, users can allocate resources more efficiently, contributing to personal financial well-being.

This platform extends far beyond mere resource-sharing; it is a catalyst for community building and knowledge exchange. By connecting students, faculty, and local businesses, the university experience is enriched. The bonds formed within these academic communities are strengthened as individuals collaborate, sharing not only equipment but also their expertise and experiences. The Equipment Lending and Rental System is a powerful tool for fostering collective responsibility and unity.

Our system is designed with trust and security as paramount concerns. Users must be confident in the reliability and quality of the equipment they rent. Meanwhile, businesses need assurance that their assets are handled with care. A robust feedback system and meticulous quality control processes ensure that these concerns are addressed, creating a foundation of trust that underpins the entire platform.

As this report highlights, the potential of the Equipment Lending and Rental System is boundless. Beyond the confines of university communities, this model can be adapted to other contexts, from neighboring schools to the wider society. Scaling and growth are essential to the platform's success, and careful planning will be required to expand and meet the evolving needs of users.

6. REFERENCES

-Bard

Large language model from Google AI, trained to generate text, translate languages, write different kinds of creative content, and answer questions in an informative way.

-ChatGPT

Large language model chatbot from OpenAI, trained to generate text, translate languages, write different kinds of creative content, and answer questions in an informative way.

-Architecture Diagram

https://www.cockroachlabs.com/blog/inventory-management-reference-architecture/