JAVA LAB PROJECT REPORT

Department Computer Science and Engineering School of Engineering and Sciences



Group Members

Pradeep Sai Teja AP21110010274

Raja Pavan Vignesh AP21110010291

Sai Karthik AP21110010310

Sai Sandeep AP21110010317

Yeswanthvenkatakumar AP21110010322

Submitted to

Dr. L. Srinivasa Rao

Subject Tutor –Java Lab (CSE 207L)

Project Title: Hostel Portal.

OBJECTIVE

The main objective of our project is to develop a user-friendly desktop application for our collage hostel which would be useful for hostel wardens and managers to take the Queries and complaints and also to issue the outing pass to the students, staff etc digitally which reduces the paper work and maintaining records.

Tools and Technologies used

		_	_	
-1	ш		_	_
-1	ш		⊨	۰

Visual Studio Code

Technologies Stack Used:

Java

Source code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
public class Portal extends JFrame implements ActionListener
{
  private JLabel usernameLabel, passwordLabel, backgroundLabel,
messageLabel;
  private JTextField usernameField;
  private JPasswordField passwordField;
  private JButton signupButton, loginButton, forgotButton;
  private File file, folder1,folder2;
  private JButton queryButton = new JButton("Query/Complaint");
  private JButton outingPassButton = new JButton("Outing");
  public java.util.Date date = new java.util.Date();
// constructor code
  public Portal()
    setTitle("User Sign Up / Login");
    setSize(700, 500);
    setResizable(false);
```

```
setLayout(null);//
  setDefaultCloseOperation(EXIT_ON_CLOSE);
  usernameLabel = new JLabel("Username:");
  passwordLabel = new JLabel("Password:");
  usernameField = new JTextField(20);
  passwordField = new JPasswordField(20);
  signupButton = new JButton("Sign Up");
  loginButton = new JButton("Login");
  messageLabel = new JLabel("");
  backgroundLabel = new JLabel(new ImageIcon("src/img.jpg"));
forgotButton = new JButton("Forgot Password");
forgotButton.addActionListener(this);
forgotButton.setBounds(390, 202, 200, 25);
add(forgotButton);
  signupButton.addActionListener(this);
  loginButton.addActionListener(this);
  usernameLabel.setForeground(Color.WHITE);
  passwordLabel.setForeground(Color.WHITE);
  messageLabel.setForeground(Color.RED);
  backgroundLabel.setBounds(0, 0, 700, 500);
  repaint();
  usernameLabel.setBounds(50, 100, 80, 25);
```

```
passwordLabel.setBounds(50, 150, 80, 25);
usernameField.setBounds(140, 100, 200, 25);
passwordField.setBounds(140, 150, 200, 25);
signupButton.setBounds(140, 200, 80, 25);
loginButton.setBounds(260, 200, 80, 25);
messageLabel.setBounds(140, 250, 300, 25);
queryButton.setBounds(260, 200, 80, 25);
outingPassButton.setBounds(260, 200, 80, 25);
add(backgroundLabel);
add(usernameLabel);
add(passwordLabel);
add(usernameField);
add(passwordField);
add(signupButton);
add(loginButton);
add(messageLabel);
add(backgroundLabel);
add(queryButton);
add(outingPassButton);
folder1 = new File("Login Info");
folder1.mkdirs();
folder2=new File ("Queries Info");
```

```
folder2.mkdirs();
    file = new File(folder1, "login.txt");
    if (file.exists())
    {
      try
      {
         BufferedReader reader = new BufferedReader(new FileReader(file));
         reader.readLine();
         reader.readLine();
         reader.close();
      }
      catch (IOException e)
      {
         e.printStackTrace();
      }
    }
    setVisible(true);
  }//constructor code ends
//// signup button code
  public void actionPerformed(ActionEvent e)
  {
    if (e.getSource() == signupButton)
```

```
{
      String username = usernameField.getText();
      String password = new String(passwordField.getPassword());
      if (username.equals("") || password.equals(""))
      {
         messageLabel.setText("Enter your username and password");
      }
      else if (username.equals(password)){
         JOptionPane.showMessageDialog(this, "User name and password
cannot be same", "Error", JOptionPane.ERROR_MESSAGE);
      }
      else if(password.length() < 8)
      {
         boolean len=false;
         boolean hasDigit = false;
         boolean hasLowercase = false;
         boolean hasUppercase = false;
         boolean hasSpecialChar = false;
         boolean hasWhieSpace = false;
         for (int i = 0; i < password.length(); i++)</pre>
         {
           char ch = password.charAt(i);
           if (Character.isDigit(ch)) {
```

```
hasDigit = true;
           } else if (Character.isLowerCase(ch)) {
              hasLowercase = true;
           } else if (Character.isUpperCase(ch)) {
              hasUppercase = true;
           } else if (ch == '@' || ch == '#' || ch == '$' || ch == '%' || ch == '^'
|| ch == '&' || ch == '+' || ch == '=') {
             hasSpecialChar = true;
           } else if (Character.isWhitespace(ch)==true) {
             hasWhieSpace=true;
           }
         }
         if (!hasDigit || !hasLowercase || !hasUppercase || !hasSpecialChar
|| !len|| hasWhieSpace)
         {
           JOptionPane.showMessageDialog(this, "Password did not met the
required criteria", "Error", JOptionPane.ERROR_MESSAGE);
         }
       }
      else
      {
         boolean userExists = false;
         try {
```

```
BufferedReader reader = new BufferedReader(new
FileReader(file));
           String line;
           while ((line = reader.readLine()) != null)
           {
             String[] parts = line.split(",");
             if (parts.length == 2 && parts[0].equals(username)) {
                userExists = true;
                break;
             }
           }
           reader.close();
         }
         catch (IOException ex)
         {
           ex.printStackTrace();
         }
         if (userExists)
         {
           JOptionPane.showMessageDialog(this, "User already exists",
"Error", JOptionPane.ERROR MESSAGE);
         }
         else
```

```
{
           if (!file.exists())
           {
             try
             {
                file.createNewFile();
             }
             catch (IOException ex)
             {
                ex.printStackTrace();
             }
           }
           try
             BufferedWriter writer = new BufferedWriter(new
FileWriter(file, true));
             writer.write(username + "," + password);
             writer.newLine();
             writer.close();
           }
           catch (IOException ex)
           {
             ex.printStackTrace();
```

```
JOptionPane.showMessageDialog(this, "Account created
successfully");
        }
      }
  }
  else if (e.getSource() == forgotButton)
  {
    JFrame forgotFrame = new JFrame("Forgot Password");
  forgotFrame.setDefaultCloseOperation(DISPOSE ON CLOSE);
  forgotFrame.setSize(300, 150);
  forgotFrame.setLayout(new GridLayout(3, 2));
 JLabel usernameLabel = new JLabel("Enter Username:");
 JButton submitButton = new JButton("Submit");
submitButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
  String username = usernameField.getText();
    String newPassword = JOptionPane.showInputDialog(null, "Enter new
password:");
    try {
```

}

```
BufferedReader reader = new BufferedReader(new FileReader("Login
Info/login.txt"));
      String line = reader.readLine();
       StringBuilder sb = new StringBuilder();
       while (line != null) {
         String[] loginInfo = line.split(",");
         if (loginInfo[0].equals(username)) {
           sb.append(username + "," + newPassword + "\n");
         } else {
           sb.append(line + "\n");
         }
         line = reader.readLine();
      }
      reader.close();
      BufferedWriter writer = new BufferedWriter(new FileWriter("Login
Info/login.txt"));
      writer.write(sb.toString());
      writer.close();
      JOptionPane.showMessageDialog(null, "Password changed
successfully");
    } catch (IOException ex) {
      ex.printStackTrace();
    }
  }
```

```
});
```

```
forgotFrame.add(usernameLabel);
  forgotFrame.add(usernameField);
  forgotFrame.add(new JLabel());
  forgotFrame.add(submitButton);
  forgotFrame.setVisible(true);
  }
//login button code
    else if (e.getSource() == loginButton)
    {
      String username = usernameField.getText();
      String password = new String(passwordField.getPassword());
      if (username.equals("") || password.equals(""))
      {
        messageLabel.setText("Please enter a username and password");
      }
      else
      {
        try
        {
           BufferedReader reader = new BufferedReader(new
FileReader(file));
```

```
String line;
           boolean found = false;
           while ((line = reader.readLine()) != null)
           {
             String[] parts = line.split(",");
             if (parts.length == 2 && parts[0].equals(username) &&
parts[1].equals(password))
             {
               found = true;
               break;
             }
           }
           reader.close();
           if (found)
           {
             JOptionPane.showMessageDialog(this, "Login successful");
            JFrame newWindow = new JFrame("Choose your option");
  JPanel buttonPanel = new JPanel();
  JButton queryButton = new JButton("Query/Complaint");
  JButton outingButton = new JButton("Outing");
  buttonPanel.add(outingButton);
  buttonPanel.add(queryButton);
```

```
newWindow.setSize(350, 200);
  newWindow.add(buttonPanel);
  newWindow.setVisible(true);
  usernameField.setText("");
  passwordField.setText("");
queryButton.addActionListener(new ActionListener()
  {
  public void actionPerformed(ActionEvent e)
  {
    JFrame frame = new JFrame("Query/Complaint Box");
             JTextArea textArea = new JTextArea();
             JScrollPane scrollPane = new JScrollPane(textArea);
             frame.getContentPane().add(scrollPane);
             frame.setSize(400, 300);
             frame.setVisible(true);
             frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
             File textFile = new File(folder2,username + ".txt");
             if (textFile.exists())
             {
               try
               {
```

```
BufferedReader textReader = new BufferedReader(new
FileReader(textFile));
                 String text = "";
                 String lineText;
                 while ((lineText = textReader.readLine()) != null) {
                    text += lineText + "\n";
                 }
                 textReader.close();
                 textArea.setText(text);
               }
               catch (IOException ex)
               {
                 ex.printStackTrace();
               }
             }
             frame.addWindowListener(new WindowAdapter()
             {
               public void windowClosing(WindowEvent e)
               {
```

try

{

```
BufferedWriter textWriter = new BufferedWriter(new
FileWriter(textFile));
                    textWriter.write(textArea.getText()+" ["+date+"]
"+"\n");
                    textWriter.close();
                 }
                 catch (IOException ex)
                 {
                    ex.printStackTrace();
                 }
               }
             });
  }
});
outingButton.addActionListener(new ActionListener()
{
  public void actionPerformed(ActionEvent e)
  {
    JFrame form = new JFrame("Outing Form");
    form.setSize(500, 400);
    JPanel labelPanel = new JPanel(new GridLayout(8, 1));
```

```
JPanel fieldPanel = new JPanel(new GridLayout(8, 1));
JLabel nameLabel = new JLabel("Name:");
labelPanel.add(nameLabel);
JLabel rollNumberLabel = new JLabel("Roll Number:");
labelPanel.add(rollNumberLabel);
JLabel roomNumberLabel = new JLabel("Room Number:");
labelPanel.add(roomNumberLabel);
JLabel towerNameLabel = new JLabel("Tower Name:");
labelPanel.add(towerNameLabel);
JLabel entryTimeLabel = new JLabel("Entry Time:");
labelPanel.add(entryTimeLabel);
JLabel dateOfEntryLabel = new JLabel("Date of Entry:");
labelPanel.add(dateOfEntryLabel);
  JLabel toDateLabel = new JLabel("To:");
```

```
labelPanel.add(toDateLabel);
JLabel dateOfReturnLabel = new JLabel("Date of Return:");
labelPanel.add(dateOfReturnLabel);
JTextField nameField = new JTextField();
fieldPanel.add(nameField);
JTextField rollNumberField = new JTextField();
fieldPanel.add(rollNumberField);
JTextField roomNumberField = new JTextField();
fieldPanel.add(roomNumberField);
JTextField towerNameField = new JTextField();
fieldPanel.add(towerNameField);
JTextField entryTimeField = new JTextField();
fieldPanel.add(entryTimeField);
JTextField dateOfEntryField = new JTextField();
fieldPanel.add(dateOfEntryField);
```

```
JTextField toDateField = new JTextField();
fieldPanel.add(toDateField);
JTextField dateOfReturnField = new JTextField();
fieldPanel.add(dateOfReturnField);
form.add(labelPanel, BorderLayout.WEST);
form.add(fieldPanel, BorderLayout.CENTER);
JButton saveButton = new JButton("Save");
saveButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    String name = nameField.getText();
    String rollNumber = rollNumberField.getText();
    String roomNumber = roomNumberField.getText();
    String towerName = towerNameField.getText();
    String entryTime = entryTimeField.getText();
    String dateOfEntry = dateOfEntryField.getText();
    String toDate = toDateField.getText();
    String dateOfReturn = dateOfReturnField.getText();
```

```
try {
  String folder3 = "Outing Info";
  File folder = new File(folder3);
  if (!folder.exists())
  {
    folder.mkdir();
  }
  FileWriter writer = new FileWriter(folder3 + "/" + username +
"_outing_data.txt", true);
  writer.write(date+"\n"+"Name:"+name + "\n" +"Roll Number
:"+rollNumber + "\n" +"Room Number :"+roomNumber + "\n" +"Tower
Name:"+towerName + "\n" +"Entry Time:"+entryTime + "\n" +"Date of
Entry :"+dateOfEntry + "\n" +"To :"+toDate + "\n" +"Date of Return
:"+dateOfReturn+"\n\n");
  writer.close();
  JOptionPane.showMessageDialog(null, "Data saved successfully!");
} catch (IOException ex)
{
  JOptionPane.showMessageDialog(null, "Error saving data!");
}
```

```
nameField.setText("");
         rollNumberField.setText("");
         roomNumberField.setText("");
         towerNameField.setText("");
         entryTimeField.setText("");
         dateOfEntryField.setText("");
         toDateField.setText("");
         dateOfReturnField.setText("");
      }
    });
    form.add(labelPanel, BorderLayout.WEST);
    form.add(fieldPanel, BorderLayout.CENTER);
    form.add(saveButton, BorderLayout.SOUTH);
    form.setVisible(true);
  }
});
else
```

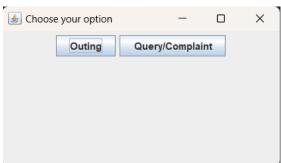
}

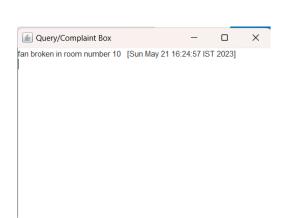
{

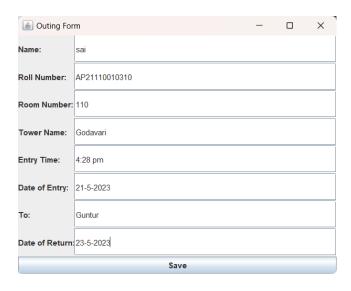
```
JOptionPane.showMessageDialog(this, "Incorrect credentials", "Error",
JOptionPane.ERROR_MESSAGE);
}
}
catch (IOException ex)
{
ex.printStackTrace();
}
}
}
}
public static void main(String args[])
{
new Portal();
}
}
```

Sample outputs









Future work

In future we will try to improve the user interface of the Application to make it more user-friendly and visually appealing. We will also consider adding interactive elements, such as draggable and resizable windows, to provide a more intuitive experience for users.

We would like to extend the functionality of the Hostel Portal to support multiple features such as Attendance food and parcel delivery.

Reference

https://github.com/rajapavan010/Hostel Portal java