

**IN
PARTNERSHIP
WITH
PLYMOUTH
UNIVERSITY**

Name: HCI, Computer Graphics, and Visualisation

Student Reference Number:

Module Code: PUSL3122

Module Name: Computer Graphics, and Visualisation

Coursework Title: Cozy Comfort Furniture (Group 55)

Deadline Date: 10, April, 2024

Member of staff responsible for coursework: Dr Taimur Bakshi

Programme: BSc (Hons) Software Engineering

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

1. 10817968 - Dissanayaka Jayawardhana
2. 10819579 - Jayasinghe A Jayasinghe
3. 10820761 - Dehiwala Nawodani
4. 10818082 – Kurugala A Jayawardhana
5. 10818178 - Warnakulasooriya Fernando
6. 10749942 – Narasinghe Thasmina

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:



Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed:

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software.....

Overall mark _____ % Assessors Initials _____ Date _____

Contents

Introduction.....	1
Background	2
Data Gathering.....	3
Design	4
Requirements.....	4
Personas	4
User Stories	5
StoryBoard	6
Low-fidelity prototype	7
high- fidelity prototype and UI	8
Implementation.....	16
Revalution	46
Summery	47
References	48

1. Introduction

Cozy Comfort Furniture introduces you to our new desktop application designed to simplify the furniture shopping experience. Our aim is straightforward to empower users like you to effortlessly design and visualize furniture for your living spaces.

At Cozy Comfort Furniture, we understand the significance of creating a comfortable and inviting home environment. We've developed an intuitive app that puts the power of furniture design at your fingertips. Whether you're a homeowner looking to furnish your space or a designer seeking to bring your vision to life, our application is tailored to meet your needs. In the following pages, we'll guide you through the journey of Cozy Comfort Furniture, offering insights into its conception, development, and functionality. We'll delve into the background and context behind our application, shedding light on the challenges it addresses and the opportunities it presents. Additionally, we'll provide a detailed overview of the methodologies employed in its creation, including user research, design considerations, and implementation strategies.

For access to the source code repository and a demonstration of the Cozy Comfort Furniture application, please refer to the following links:

GitHub Repository: <https://github.com/Plymouth-University/main-assessment-group-55.git>

YouTube Video: https://youtu.be/Q0bQRa_SNmA

2. Background

Cozy Comfort Furniture is a user-friendly desktop application tailored for individuals eager to transform their living spaces with custom furniture arrangements. By providing intuitive tools for visualization and design, the app empowers users to create personalized layouts based on specific room dimensions, shapes, and colour schemes. Whether you're a homeowner looking to revamp your living room or an interior designer seeking to craft bespoke solutions for clients, Cozy Comfort Furniture offers a seamless platform to bring your furniture ideas to life.

With Cozy Comfort Furniture, users can easily experiment with different furniture configurations in a virtual 2D environment, gaining clarity on how each piece will fit and interact within the room. Moreover, the application elevates the design experience by immersing users in lifelike 3D renderings, offering a realistic portrayal of the proposed furniture arrangements. This comprehensive visualization capability enables users to make informed decisions about furniture placement and aesthetics, ensuring optimal spatial harmony and functionality.

Designed with simplicity and practicality in mind, Cozy Comfort Furniture caters to a diverse range of users, including homeowners looking to optimize their living spaces, interior designers seeking to streamline the design process, and furniture enthusiasts eager to unleash their creativity. By providing a user-friendly interface and robust functionality, the application empowers users to curate inviting and visually appealing environments that reflect their individual style and preferences.

3. Data gathering

To gather additional requirements for our product, we engaged in various methods to ensure a comprehensive understanding of user needs. Firstly, we conducted surveys among our target audience to gather insights into their preferences and pain points regarding furniture shopping and design. Additionally, we conducted interviews with potential users, including homeowners and interior designers, to gain deeper insights into their specific requirements and expectations from the application.

Furthermore, we analysed feedback and reviews from existing furniture design tools and similar applications to identify common features and functionalities that users find valuable. This comparative analysis helped us prioritize features and refine our product roadmap to better meet user expectations. Additionally, we utilized user stories and personas to create fictional characters representing different user demographics and scenarios, allowing us to empathize with users' needs and tailor our product accordingly.

Once the data was collected, we employed qualitative analysis techniques to identify recurring themes, patterns, and insights from the gathered information. This involved categorizing and synthesizing the data to extract key requirements and user preferences. We then prioritized these requirements based on their relevance and impact on the user experience, informing the design and development process of our product. Overall, our data gathering approach ensured that our product is informed by user insights and designed to effectively address their needs and preferences.

4. Design

❖ Requirements

- Customers can go to the register page & login page.
- Checking dashboard, can choose a design user favourite.
- Customers can create a new design, based on the room size, shape and colour scheme.
- A design visualises in 2D.
- A design visualises in 3D.
- Scale the design to best fit the room based on the size.
- Customer can see shading in the design parts.
- Change the colour of the design parts.
- Customer can save the design.
- Can add, edit and delete the design.
- Customer can view the orders.
- Customer can check the privacy policy, terms & conditions, about us in the settings.

❖ Persons

Persona 1: Sarah

- Background: Sarah is a 35-year-old interior designer who loves making rooms look amazing. She's been doing this for a long time and is always looking for easier ways to do her job.
- Goals: Sarah wants to use an easy computer program to see how furniture will look in her clients' rooms before she buys anything. She wants to try out different colours and sizes to see what works best.
- Needs: Sarah needs a simple computer program where she can put in the size of the room and pick furniture from a big list. She wants to be able to change the size and colour of the furniture and see how it looks in the room. She also wants to save her designs to show her clients.

Persona 2: David

- Background: David is a 42-year-old guy who just got a new house. He wants to find furniture that looks good and fits in his new place without having to go to a store.
- Goals: David wants to use a computer program to see how different furniture will look in his rooms before he buys anything. He wants to see what it looks like in 3D and try different colours and sizes.
- Needs: David needs a simple computer program where he can pick furniture like chairs and tables from a list. He wants to be able to change the size and colour of the furniture and see how it looks in his rooms. He also wants the program to suggest other furniture that goes well with what he picks.

❖ ***User Stories***

User Story 1: As Sarah, I want to be able to put in the size of the room and pick furniture from a big list, so I can make nice designs for my clients.

User Story 2: As David, I want to see how furniture will look in my rooms before I buy anything, so I can make sure it looks good.

User Story 3: As Sarah, I want to change the size and colour of the furniture and save my designs, so I can show my clients different options.

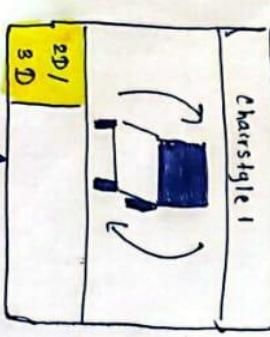
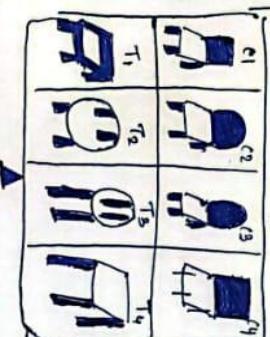
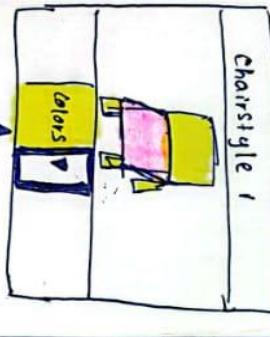
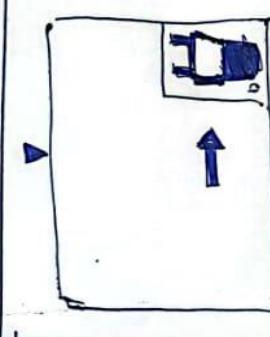
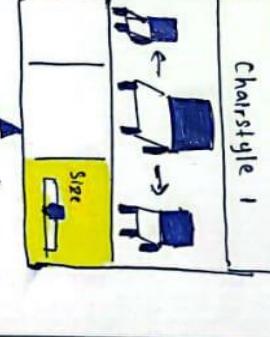
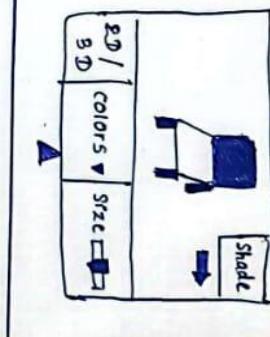
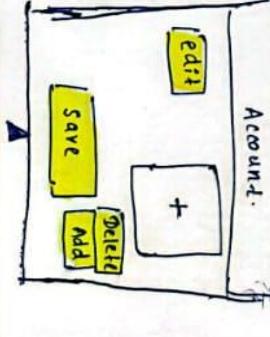
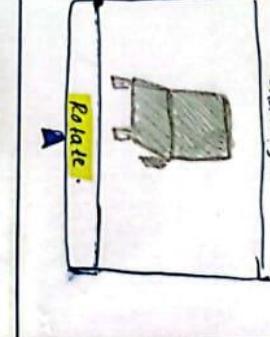
User Story 4: As David, I want to compare different furniture options and see which one looks best in my rooms, so I can make a good choice.

User Story 5: As Sarah, I want the program to be easy to use and not take too long to make designs, so I can get my work done quickly.

User Story 6: As David, I want the program to suggest other furniture that goes well with what I pick, so I can create a nice-looking room.

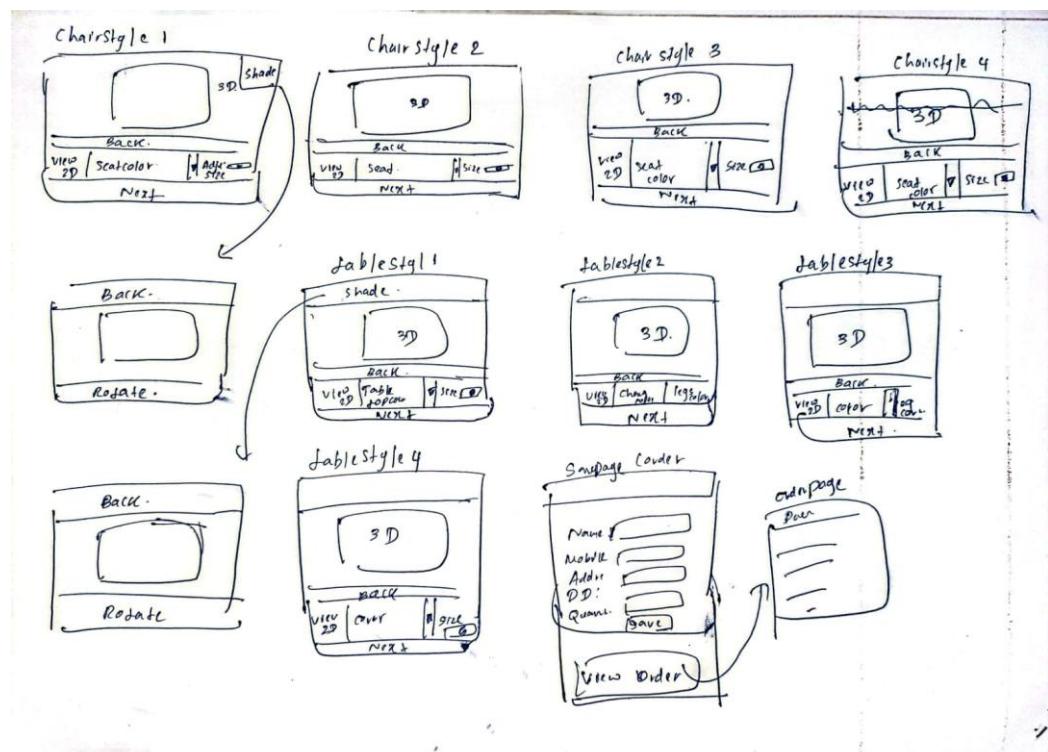
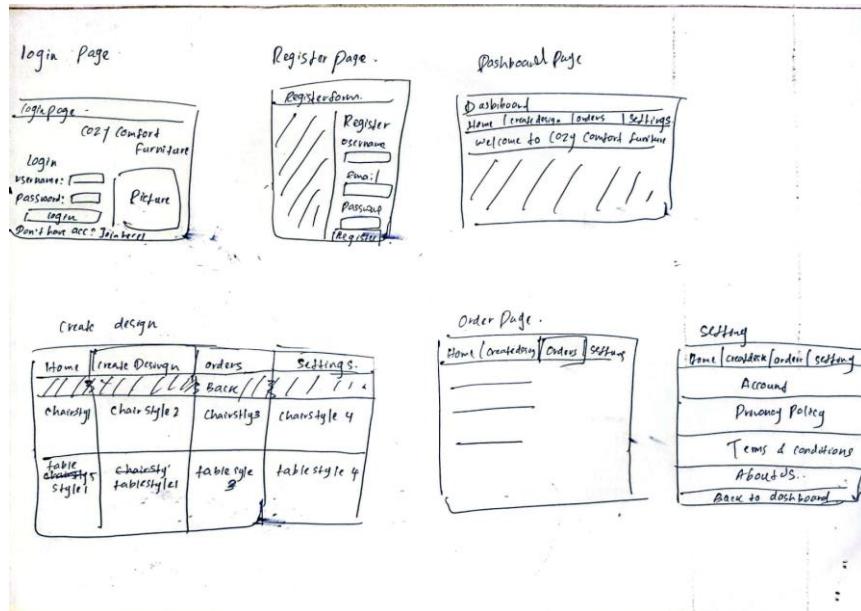
User Story 7: As Sarah, I want to be able to share my designs with my clients and get their feedback, so I can make changes if they want something different.

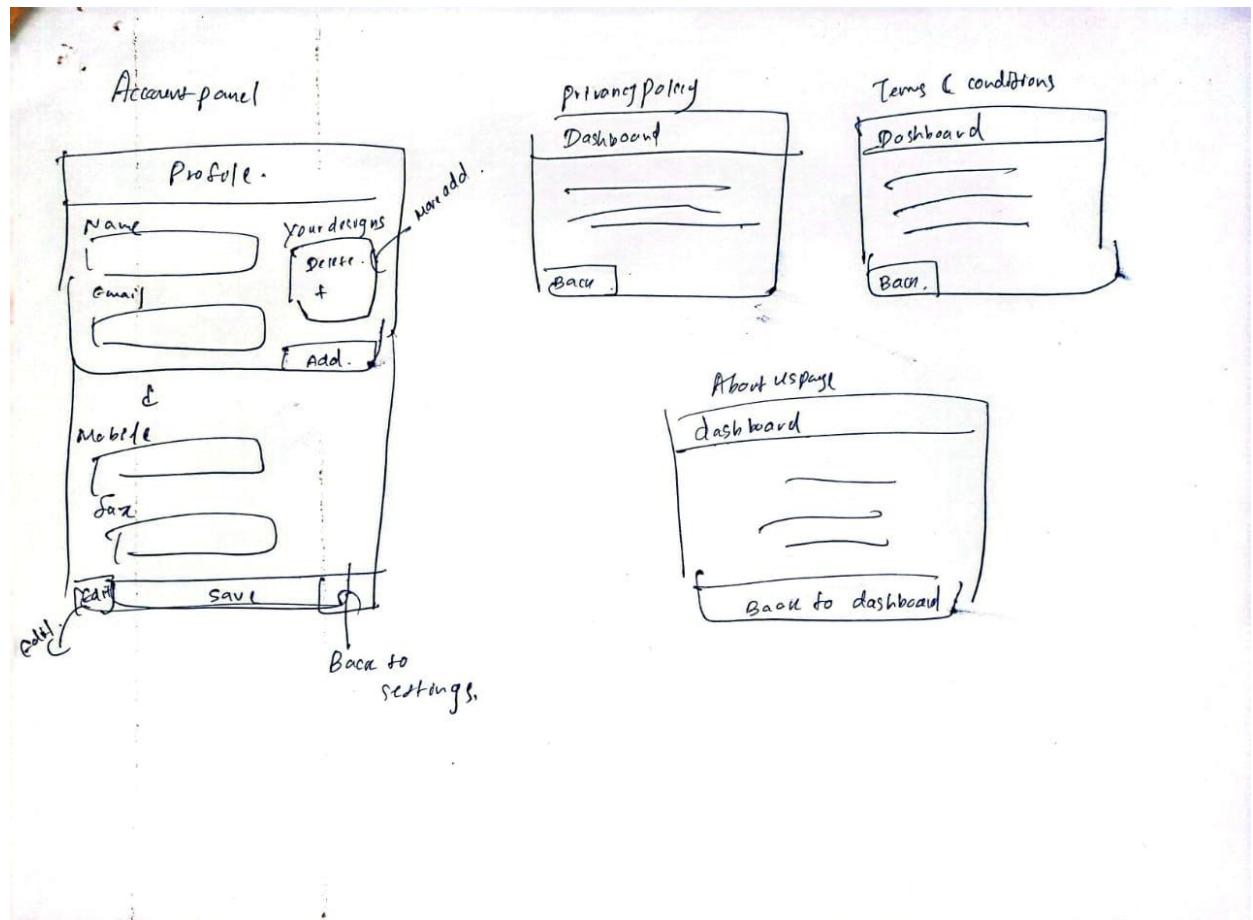
❖ Story board

<p>create design.</p> 	<p>create design</p> 
<p>charstyle 1</p> 	<p>charstyle 1</p> 
<p>charstyle 1</p> 	<p>charstyle 1</p> 
<p>charstyle 1</p> 	<p>charstyle 1</p> 

❖ Low-fidelity prototype

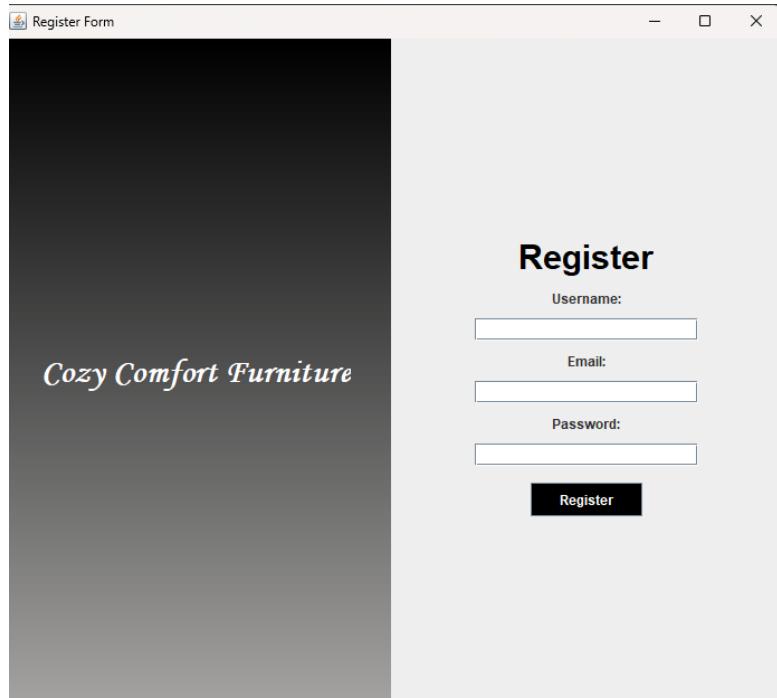
A low-fidelity prototype can be created using simple tools like pen and paper. With this approach, designers can quickly sketch out the basic layout, screens, and interactions of the application. By using pen and paper, they can iterate rapidly, gather feedback, and make necessary adjustments to the design before moving to higher-fidelity prototypes.





❖ High-fidelity prototypes with UI





The screenshot shows a dashboard window titled "Dashboard". The top navigation bar includes "Home", "Create Design", "Orders", and "Settings". Below the navigation, there are two rows of four items each, labeled "Chair Style 1" through "Chair Style 4" and "Table Style 1" through "Table Style 4". Each item has a small image and a brief description. For example, Chair Style 1 is described as the newest model for dining rooms, while Table Style 4 is described as having a unique antique-style design.

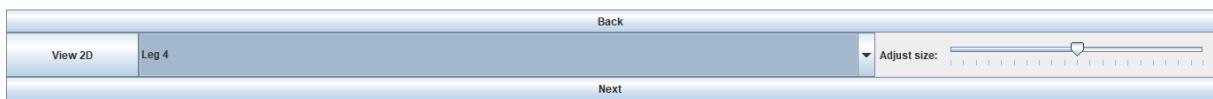
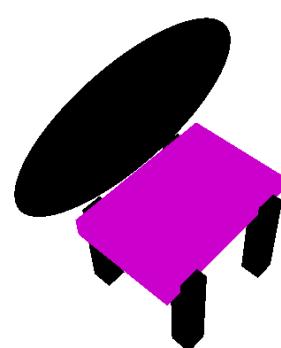
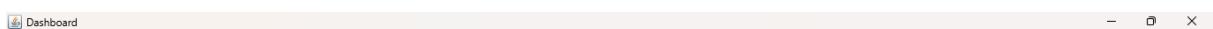
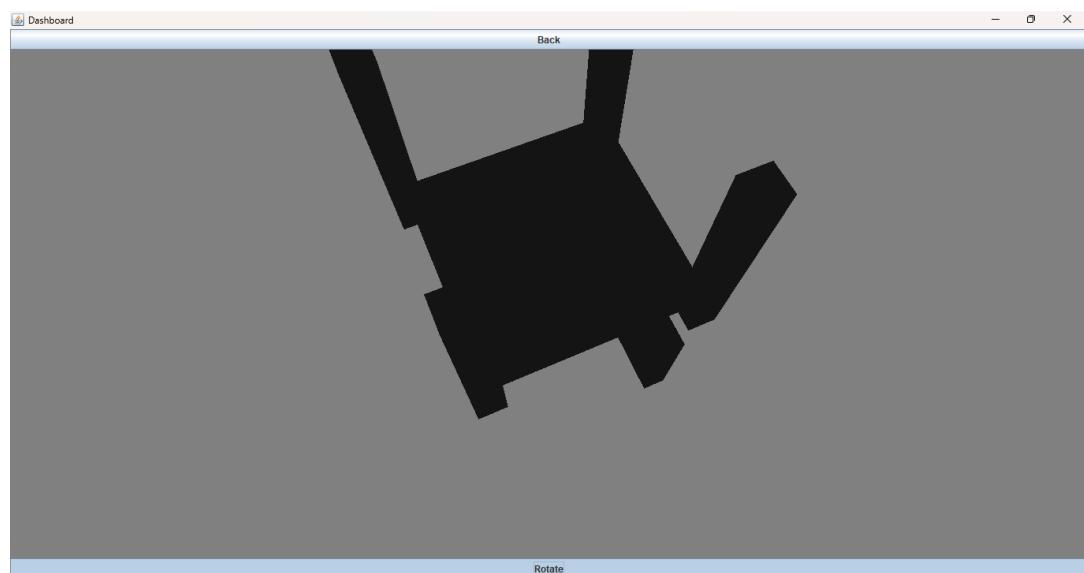
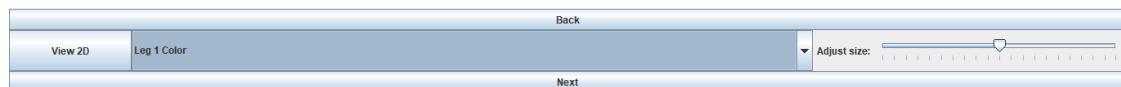
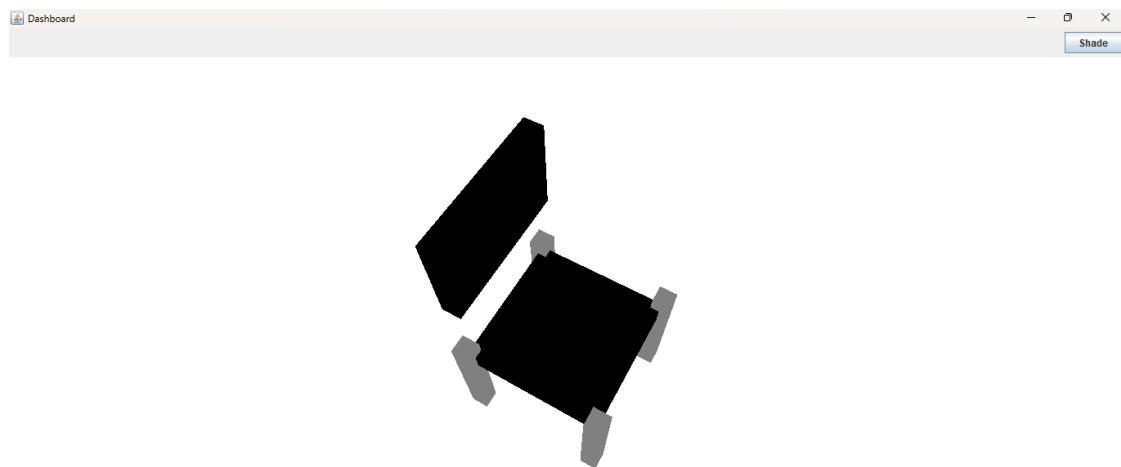
Chair Style 1	Chair Style 2	Chair Style 3	Chair Style 4
This is the newest model for your dining room, adding a touch of elegance. Customize it to make it your own.	Explore our collection of rounded tables, featuring various models perfect for your living room.	This rounded chair offers elegance and versatility, ideal for enhancing the ambiance of both your dining and living rooms.	This antique-style chair boasts a unique design, adding a touch of character to your living room decor.

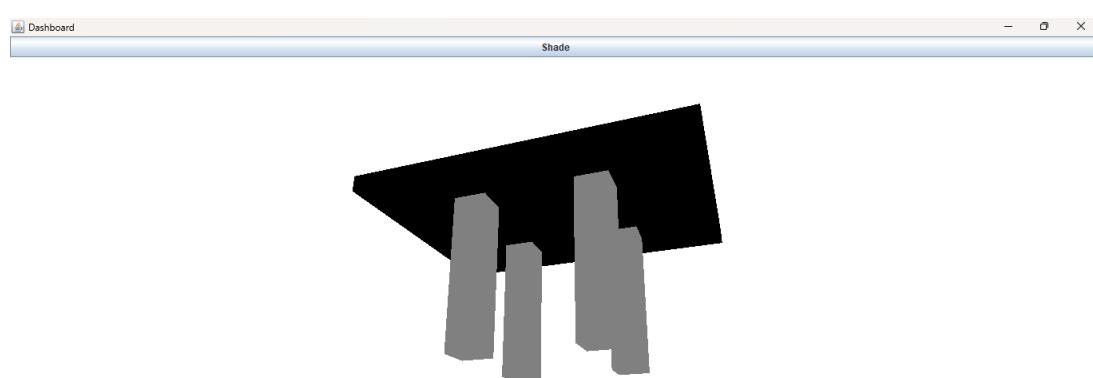
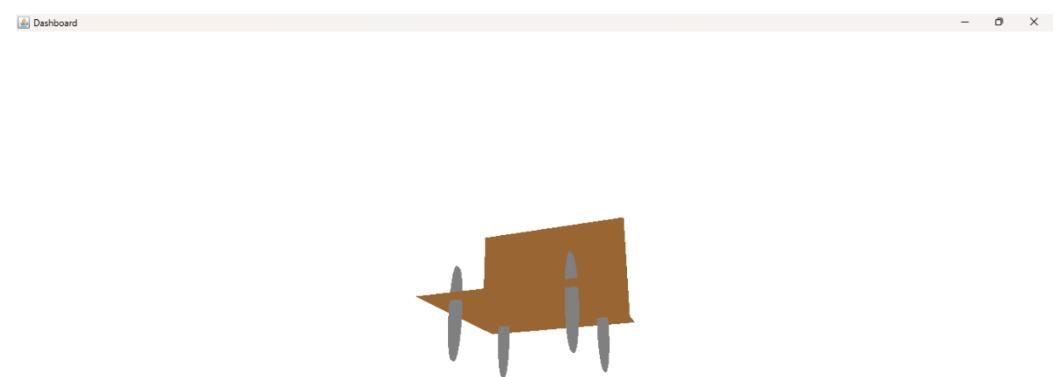
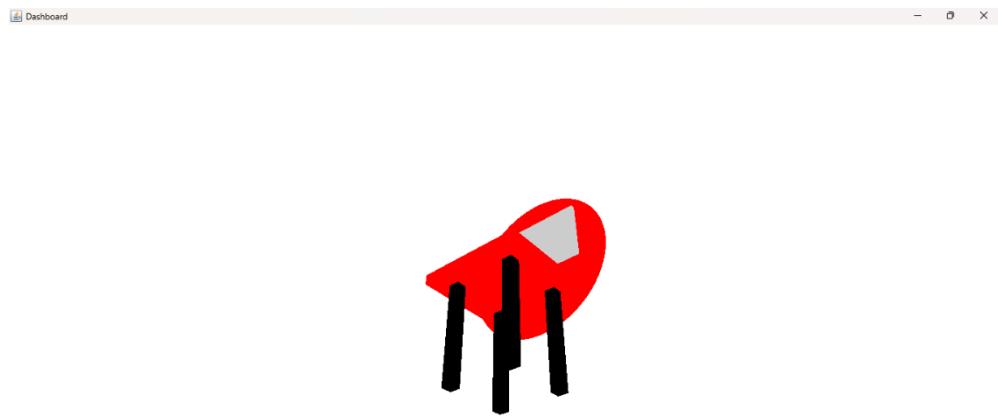
Table Style 1	Table Style 2	Table Style 3	Table Style 4
This elegant and spacious table is perfect for your dining room, adding a touch of grandeur to your space.	This table features a sleek design with four sturdy legs and a glass top, elegantly framed in rounded wood.	This table is suitable for outdoor gardens and adds the best touch to your garden.	This versatile table serves as the perfect option for a study desk or a coffee table, offering both functionality and style.

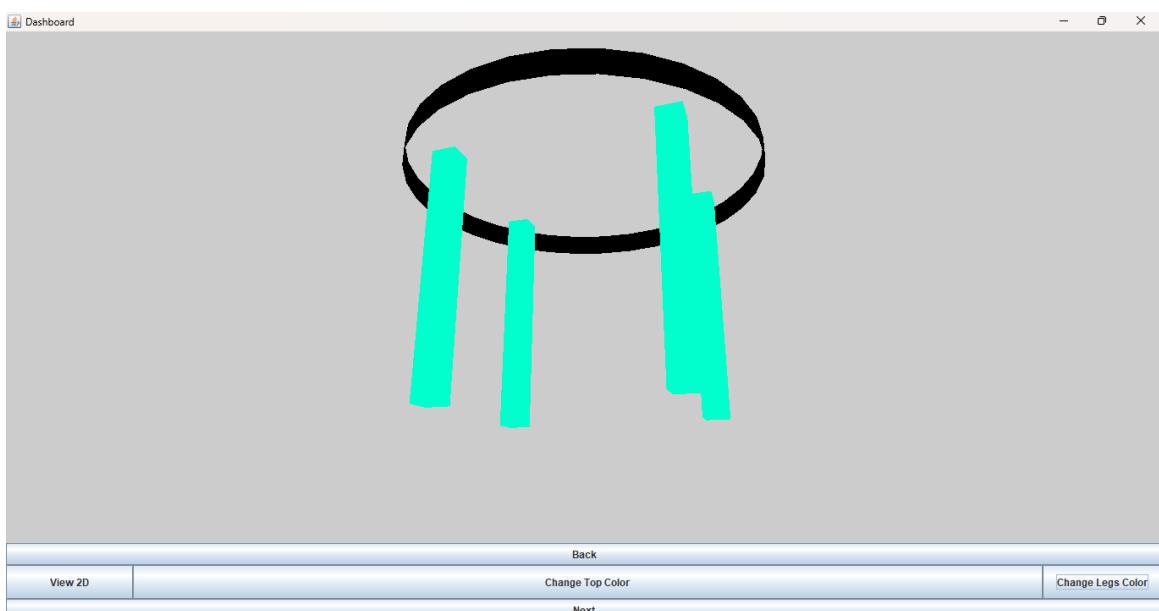
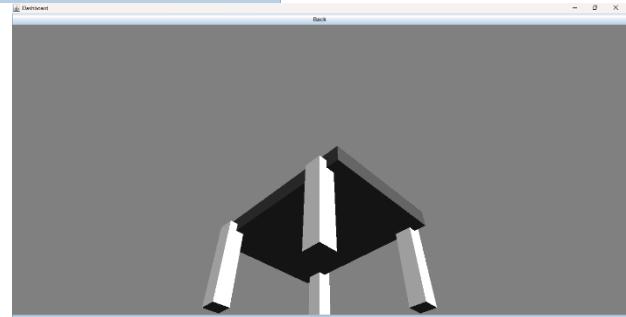
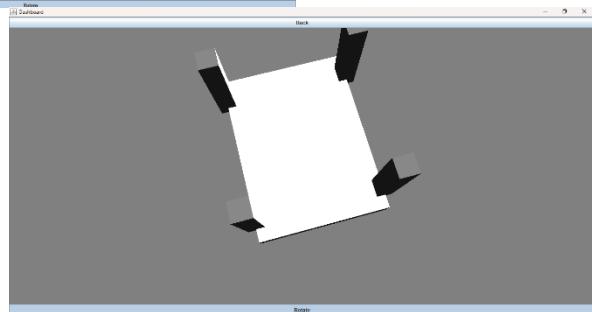
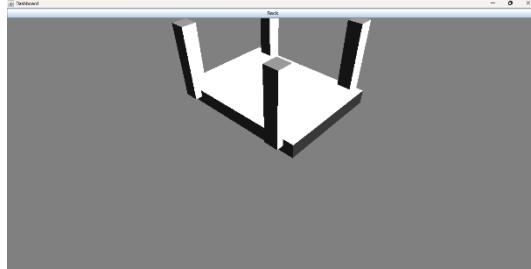
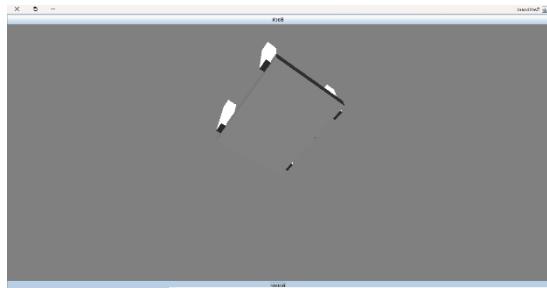
The screenshot shows the "Orders" section of the dashboard. The top navigation bar includes "Home", "Create Design", "Orders", and "Settings". Below the navigation, a message displays five recent order IDs and their expected delivery dates. At the bottom of the page is a "Back to Home" link.

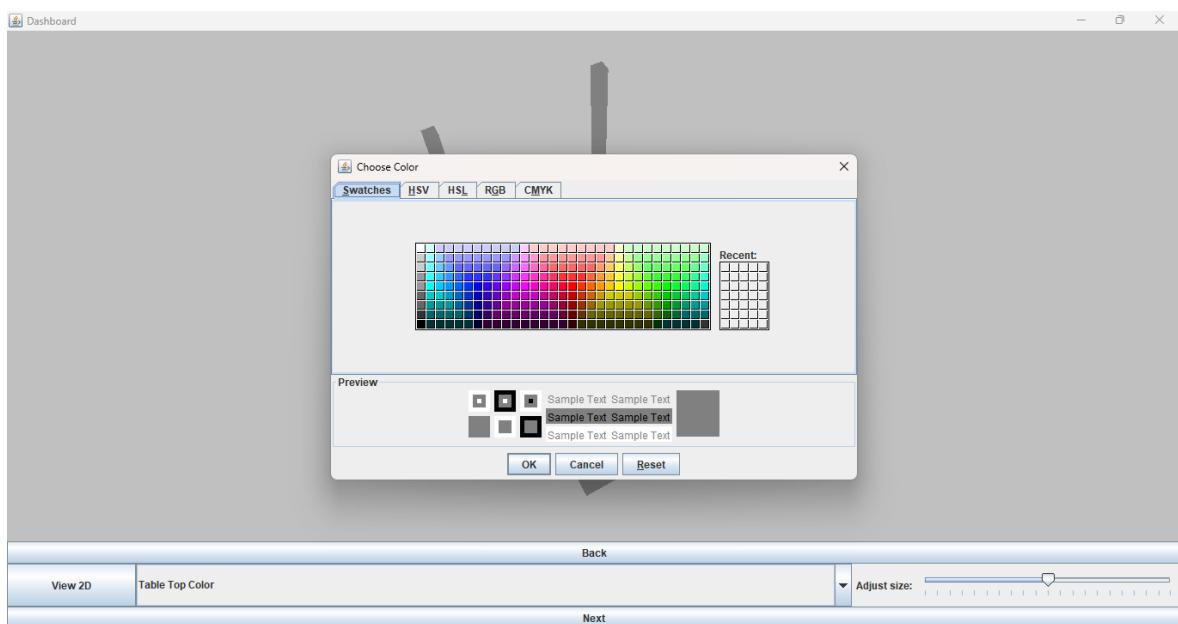
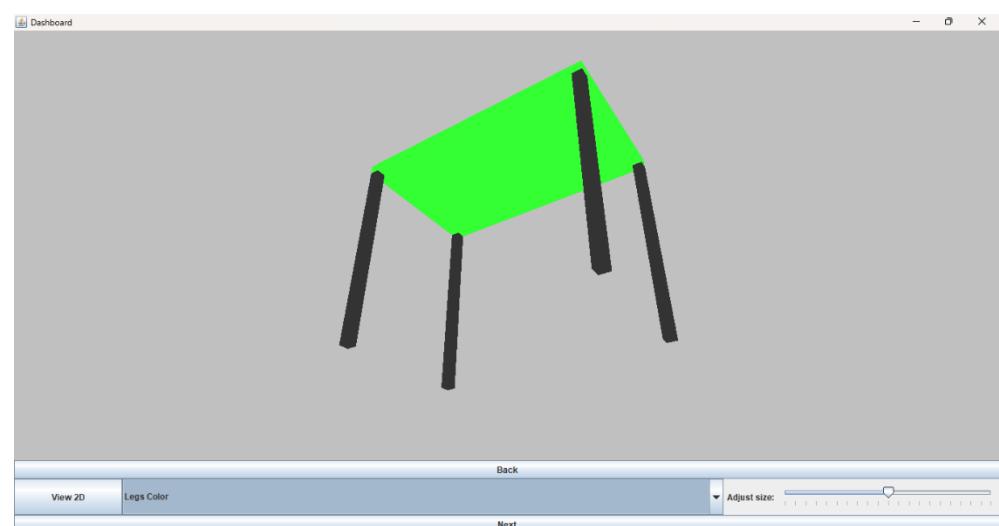
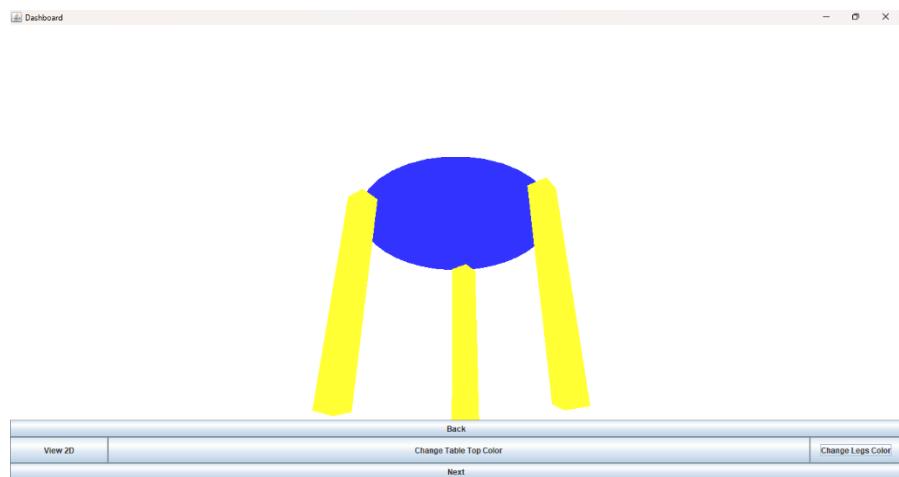
Order ID: ORD1712609885582, Expected Delivery Date: wataraka
Order ID: ORD1712611473635, Expected Delivery Date: 2024/05/18
Order ID: ORD1712612353767, Expected Delivery Date: 2024/6/12
Order ID: ORD1712614333548, Expected Delivery Date: 5/6

[Back to Home](#)









Dashboard

Name:	<input type="text"/>
Mobile:	<input type="text"/>
Address:	<input type="text"/>
Expected Delivery Date:	<input type="text"/>
Quantity:	<input type="text"/>

[View Order](#)

Dashboard

- [Home](#)
- [Create Design](#)
- [Orders](#)
- [Settings](#)

Settings

Account

Privacy Policy

Terms and Conditions

About Us

[Back to Dashboard](#)

Dashboard

Profile

Your Designs

<input type="button" value="Delete"/>	<input type="button" value="Delete"/>	<input type="button" value="Delete"/>
+	+	+

Look In: ドキュメント

- Custom Office Templates
- film n tv serise
- GitHub
- MATLAB
- ShareX
- Visual Studio 2022
- WindowsPowerShell
- Zoom
- AI AND ML (AutoRecovered).pdf
- AI AND ML.pdf
- fullstack video demo.docx
- fullstack_video_demo(group 29).docx

File Name:

Files of Type: All Files

Name

Email

Mobile

Fax

[Back to Settings](#)

This Privacy Policy explains how FurShop ("we," "us," "our") collects, uses, shares, and protects information in relation to our mobile application ("App") and website (collectively, the "Service"), and your choices about the collection and use of your information.

1. Information Collection and Use
We may collect and use personal information to provide and improve our Service. This may include information such as your name, email address, and phone number when you register for an account, use our contact form, or make a purchase.

2. Log Data
We may also collect information that your browser sends whenever you visit our Service ("Log Data"). This Log Data may include information such as your computer's Internet Protocol ("IP") address, browser type, browser version, the pages of our Service that you visit, the time and date of your visit, the time spent on those pages, and other statistics.

3. Cookies
Cookies are files with a small amount of data, which may include an anonymous unique identifier. Cookies are sent to your browser from a website and stored on your computer's hard drive. We use "cookies" to collect information and improve our Service. You can instruct your browser to refuse all cookies or to indicate when a cookie is being sent. However, if you do not accept cookies, you may not be able to use some portions of our Service.

4. Security
The security of your personal information is important to us, but remember that no method of transmission over the Internet or method of electronic storage is 100% secure. While we strive to use commercially acceptable means to protect your personal information, we cannot guarantee its absolute security.

5. Changes to This Privacy Policy
We may update our Privacy Policy from time to time. We will notify you of any changes by posting the new Privacy Policy on this page. You are advised to review this Privacy Policy periodically for any changes.

6. Contact Us
If you have any questions or suggestions about our Privacy Policy, do not hesitate to contact us at contact@example.com.

Back

Terms and Conditions

1. Introduction
These Terms and Conditions govern your use of [Your Company Name]'s desktop application and any related services provided therein.

2. Acceptance of Terms
By accessing or using the application, you agree to be bound by these Terms and Conditions. If you do not agree to all of these terms, please do not use the application.

3. Use of the Application
The application is provided on an "as is" and "as available" basis. [Your Company Name] reserves the right to modify or discontinue, temporarily or permanently, the application or any features or portions thereof without prior notice.

4. Intellectual Property Rights
The application and its original content, features, and functionality are owned by [Your Company Name] and are protected by international copyright, trademark, patent, trade secret, and other intellectual property or proprietary rights laws.

5. Privacy Policy
Your use of the application is also subject to our Privacy Policy. Please review our Privacy Policy, which governs the collection, use, and disclosure of personal information.

6. Limitation of Liability
In no event shall [Your Company Name] be liable for any indirect, incidental, special, consequential or punitive damages, including without limitation, loss of profits, data, use, goodwill, or other intangible losses, arising out of or in connection with your use of the application.

7. Indemnification
You agree to indemnify, defend, and hold harmless [Your Company Name], its officers, directors, employees, agents, and third parties from and against any claims, liabilities, damages, losses, and expenses, including reasonable attorneys' fees and costs, arising out of or in any way connected with your use of the application or violation of these Terms and Conditions.

8. Changes to Terms
[Your Company Name] reserves the right, at its sole discretion, to modify or replace these Terms and Conditions at any time. It is your responsibility to review these Terms and Conditions periodically for changes. Your continued use of the application following the posting of any changes to these Terms and Conditions constitutes acceptance of those changes.

9. Contact Us
If you have any questions about these Terms and Conditions, please contact us at [contact email].

Back

About Us

For over 75 years, Cozy Comfort Furniture has stood as a beacon of quality and excellence in the furniture industry. Since our inception, we've remained steadfast in our commitment to delivering unparalleled craftsmanship and exceptional service to our valued customers. Our journey has been one of continuous evolution, driven by a passion for innovation and a relentless pursuit of perfection.

At Cozy Comfort Furniture, we understand that furniture is more than just pieces arranged in a room; it's a reflection of one's lifestyle, personality, and taste. That's why we've made it our mission to not only meet but exceed the expectations of our discerning clientele. From classic designs to cutting-edge innovations, each piece is carefully curated to blend timeless elegance with modern flair.

Our success wouldn't be possible without the dedication and talent of our team. We believe in nurturing a culture of creativity, collaboration, and empowerment, providing our employees with the tools and support they need to thrive. By fostering a dynamic work environment that encourages growth and innovation, we're able to stay ahead of the curve and continue delivering exceptional results.

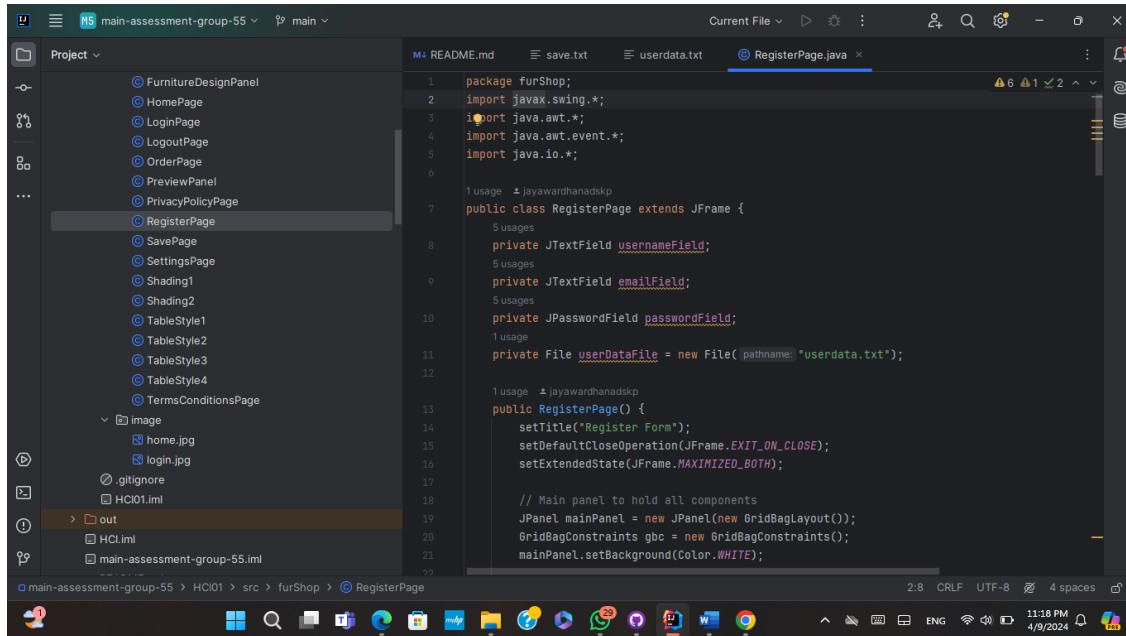
As we look to the future, we remain committed to upholding the values that have guided us since day one: integrity, quality, and customer satisfaction. By embracing change, embracing new technologies, and staying true to our core principles, we're confident that Cozy Comfort Furniture will continue to be a trusted name in the industry for generations to come.

Back

5. Implementation

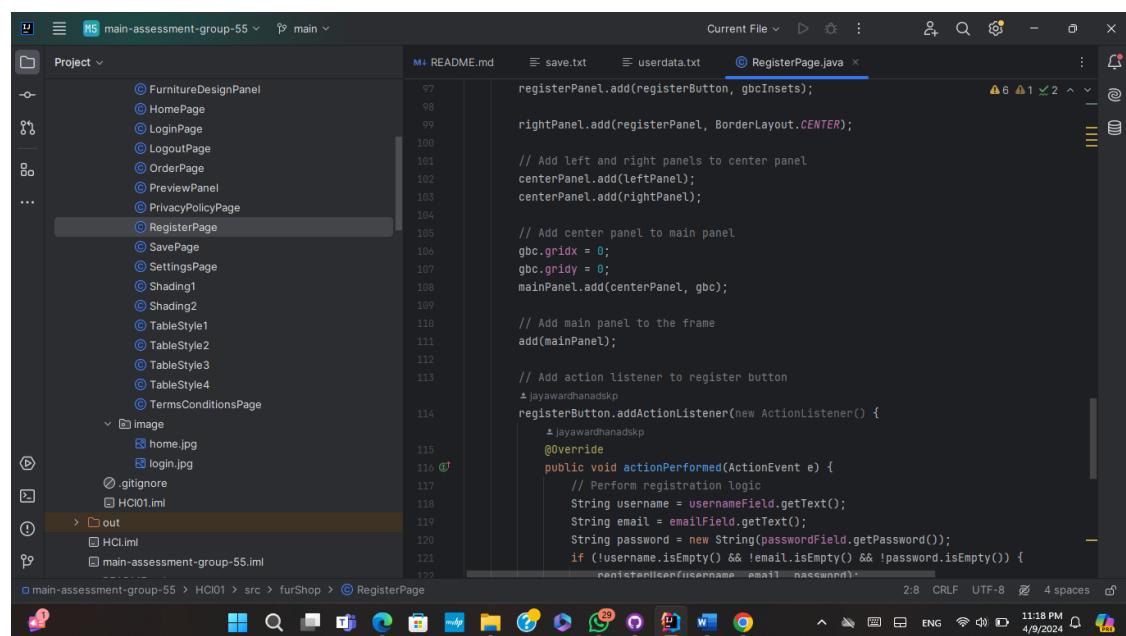
- ❖ Link to the GitHub repository: <https://github.com/Plymouth-University/main-assessment-group-55.git>

- Register java class.



The screenshot shows a Java code editor with the file `RegisterPage.java` open. The code implements a `JFrame` for user registration. It includes imports for `javax.swing.*`, `java.awt.*`, `java.awt.event.*`, and `java.io.*`. The class `RegisterPage` has a constructor that sets the title to "Register Form", sets the default close operation to `JFrame.EXIT_ON_CLOSE`, and sets the extended state to `JFrame.MAXIMIZED_BOTH`. It contains a main panel using `GridBagLayout` and adds left and right panels to it. A center panel is also added. An action listener is registered for the register button. The code uses several private fields for text fields and a password field, and a file for saving user data.

```
1 package furShop;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.io.*;
6
7 Usage ▲ jayawardhanadskp
8 public class RegisterPage extends JFrame {
9     Usages
10    private JTextField usernameField;
11    Usages
12    private JTextField emailField;
13    Usages
14    private JPasswordField passwordField;
15    Usage
16    private File userDataFile = new File( pathname, "userdata.txt");
17
18 Usage ▲ jayawardhanadskp
19 public RegisterPage() {
20     setTitle("Register Form");
21     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22     setExtendedState(JFrame.MAXIMIZED_BOTH);
23
24     // Main panel to hold all components
25     JPanel mainPanel = new JPanel(new GridBagLayout());
26     GridBagConstraints gbc = new GridBagConstraints();
27     mainPanel.setBackground(Color.WHITE);
28
29     // Add left and right panels to center panel
30     centerPanel.add(leftPanel);
31     centerPanel.add(rightPanel);
32
33     // Add center panel to main panel
34     gbc.gridx = 0;
35     gbc.gridy = 0;
36     mainPanel.add(centerPanel, gbc);
37
38     // Add main panel to the frame
39     add(mainPanel);
40
41     // Add action listener to register button
42     registerButton.addActionListener(new ActionListener() {
43         Usage ▲ jayawardhanadskp
44         @Override
45         public void actionPerformed(ActionEvent e) {
46             // Perform registration logic
47             String username = usernameField.getText();
48             String email = emailField.getText();
49             String password = new String(passwordField.getPassword());
50             if (!username.isEmpty() && !email.isEmpty() && !password.isEmpty()) {
51                 registerUser(username, email, password);
52             }
53         }
54     });
55 }
56
57 Usage ▲ jayawardhanadskp
58 private void registerUser(String username, String email, String password) {
59     try {
60         String data = username + "," + email + "," + password;
61         userDataFile.write(data);
62     } catch (IOException e) {
63         e.printStackTrace();
64     }
65 }
```



The screenshot continues the `registerUser` method from the previous code block. It reads the user input from the text fields and writes it to the `userData.txt` file. The code handles potential `IOException`.

```
66     try {
67         String data = username + "," + email + "," + password;
68         userDataFile.write(data);
69     } catch (IOException e) {
70         e.printStackTrace();
71     }
72 }
```

```
// Center panel to hold left and right panels
JPanel centerPanel = new JPanel(new GridLayout(1, cols: 2));
centerPanel.setPreferredSize(new Dimension(width: 700, height: 600));

// Left Panel with Shaded Background
JPanel leftPanel = paintComponent(g) {
    super.paintComponent(g);
    Graphics2D g2d = (Graphics2D) g.create();

    // Define the gradient colors
    Color color1 = new Color(0, 0, 0);
    Color color2 = new Color(166, 166, 166);

    // Create a vertical gradient
    GradientPaint gradient = new GradientPaint(
        x1: 0, y1: 0, color1,
        x2: 0, getheight(), color2
    );

    // Apply the gradient
    g2d.setPaint(gradient);
    g2d.fillRect(0, 0, getWidth(), getHeight());
    g2d.dispose();
};

leftPanel.setLayout(new GridBagLayout());

```

```
g2d.dispose();
};

leftPanel.setLayout(new GridBagLayout());
leftPanel.setPreferredSize(new Dimension(width: 200, height: 200));

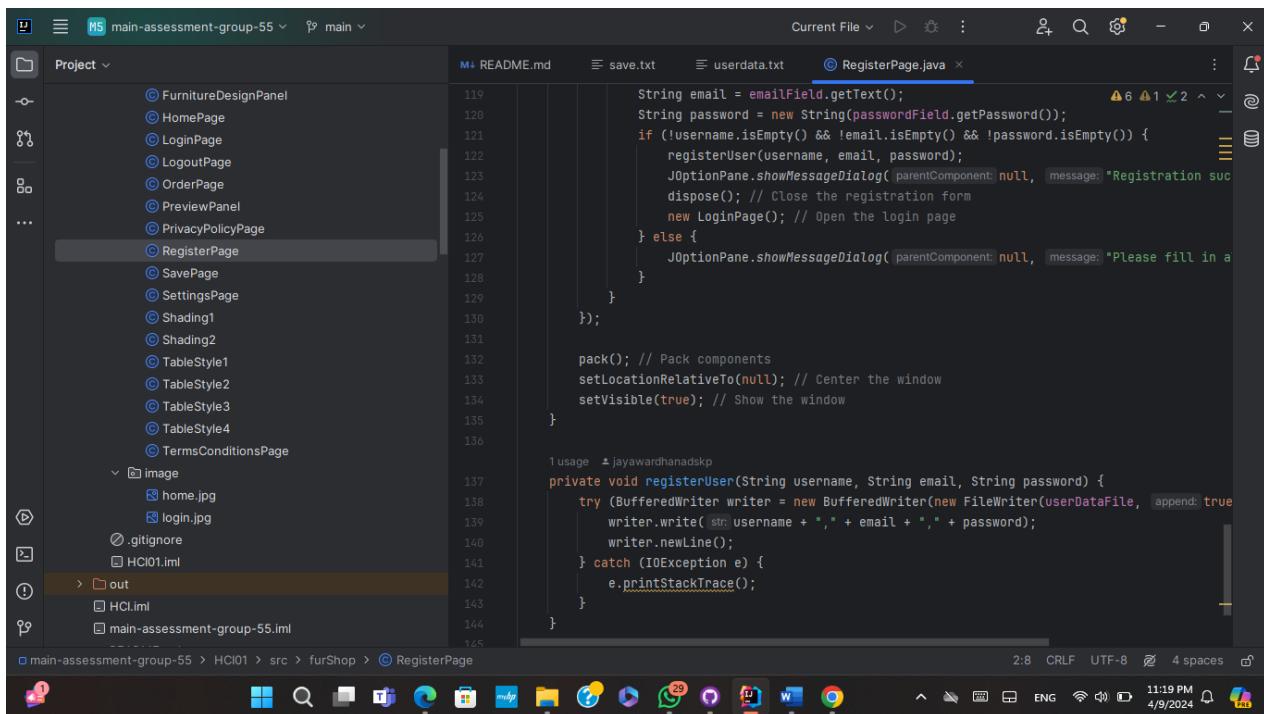
// Add label with white color text over the background
JLabel leftLabel = new JLabel(text: "Cozy Comfort Furniture");
leftLabel.setFont(new Font(name: "Monotype Corsiva", Font.BOLD, size: 30));
leftLabel.setForeground(Color.WHITE);
leftPanel.add(leftLabel, new GridBagConstraints());

// Right Panel (Register Form)
JPanel rightPanel = new JPanel(new BorderLayout());
JPanel registerPanel = new JPanel(new GridBagLayout());
registerPanel.setBorder(BorderFactory.createEmptyBorder(top: 20, left: 20, bottom: 20, right: 20));
JLabel registerLabel = new JLabel(text: "Register");
registerLabel.setForeground(new Color(0, 0, 0)); // Deep Sky Blue color
registerLabel.setFont(new Font(name: "Arial", Font.BOLD, size: 30));
usernameField = new JTextField();
usernameField.setPreferredSize(new Dimension(width: 200, height: 20)); // Adjusted width of
usernameField.setToolTipText("username");
emailField = new JTextField();
emailField.setPreferredSize(new Dimension(width: 200, height: 20)); // Adjusted width of
emailField.setToolTipText("Email");
passwordField = new JPasswordField();
passwordField.setPreferredSize(new Dimension(width: 200, height: 20)); // Adjusted width of
passwordField.setToolTipText("Password");
JButton registerButton = new JButton(text: "Register");

```

```
passwordField = new JPasswordField();
passwordField.setPreferredSize(new Dimension(width: 200, height: 20)); // Adjusted width of
passwordField.setToolTipText("Password");
JButton registerButton = new JButton(text: "Register");
registerButton.setBackground(new Color(0, 0, 0));
registerButton.setForeground(Color.WHITE);
registerButton.setPreferredSize(new Dimension(width: 100, height: 30));
GridBagConstraints gbcInsets = new GridBagConstraints();
gbcInsets.insets = new Insets(top: 5, left: 0, bottom: 5, right: 0);
registerPanel.add(registerLabel, gbcInsets);
gbcInsets.gridx = 1;
registerPanel.add(new JLabel(text: "Username:"), gbcInsets);
gbcInsets.gridx = 2;
registerPanel.add(usernameField, gbcInsets);
gbcInsets.gridx = 3;
registerPanel.add(new JLabel(text: "Email:"), gbcInsets);
gbcInsets.gridx = 4;
registerPanel.add(emailField, gbcInsets);
gbcInsets.gridx = 5;
registerPanel.add(new JLabel(text: "Password:"), gbcInsets);
gbcInsets.gridx = 6;
registerPanel.add(passwordField, gbcInsets);
gbcInsets.gridx = 7;
gbcInsets.insets = new Insets(top: 10, left: 0, bottom: 0, right: 0);
registerPanel.add(registerButton, gbcInsets);
rightPanel.add(registerPanel, BorderLayout.CENTER);

```

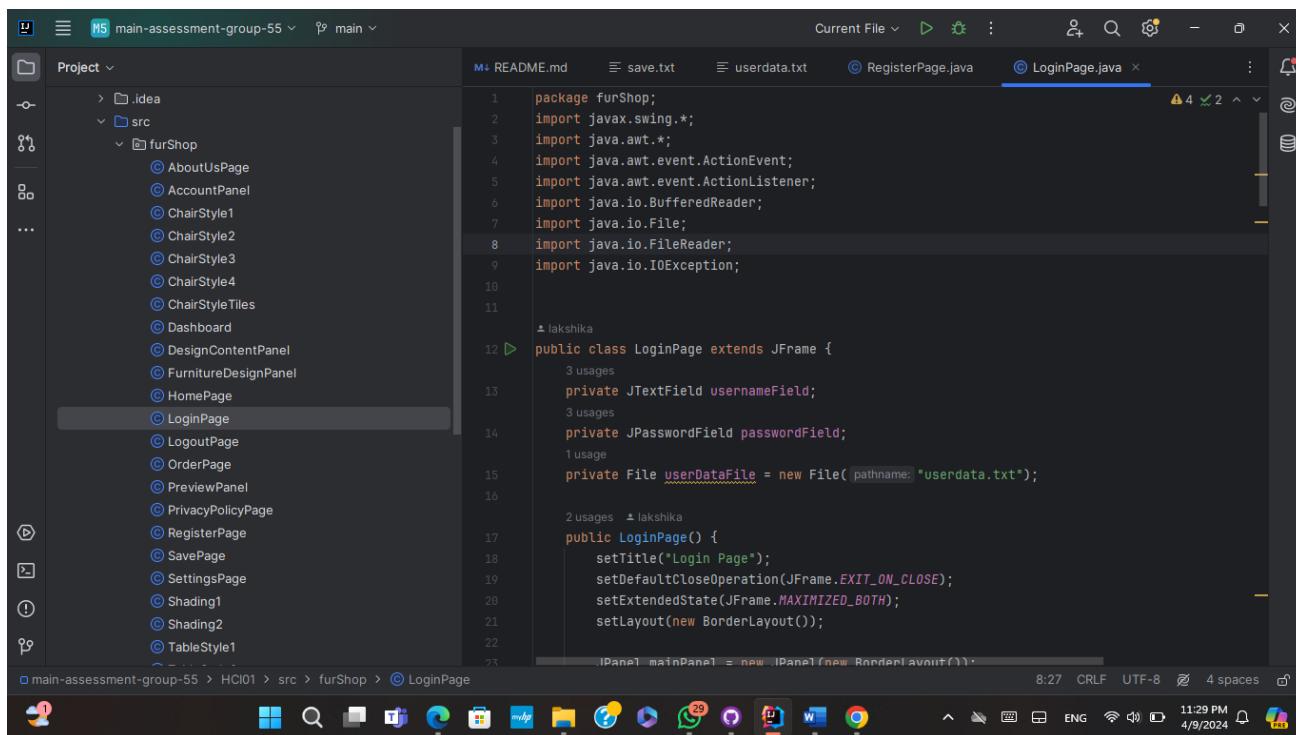


```
String email = emailField.getText();
String password = new String(passwordField.getPassword());
if (username.isEmpty() && email.isEmpty() && !password.isEmpty()) {
    registerUser(username, email, password);
    JOptionPane.showMessageDialog(parentComponent, null, message: "Registration successful");
    dispose(); // Close the registration form
    new LoginPage(); // Open the login page
} else {
    JOptionPane.showMessageDialog(parentComponent, null, message: "Please fill in all fields");
}
});

pack(); // Pack components
setLocationRelativeTo(null); // Center the window
setVisible(true); // Show the window
}

1 usage ↳ jayawardhanadskp
private void registerUser(String username, String email, String password) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(userdataFile, append: true))) {
        writer.write(username + "," + email + "," + password);
        writer.newLine();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

➤ Login java class.



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

public class LoginPage extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private File userDataFile = new File(pathname: "userdata.txt");

    public LoginPage() {
        setTitle("Login Page");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        setLayout(new BorderLayout());
        IPanel mainPanel = new IPanel(new RenderPanel());
    }
}
```

Project tree:

- .idea
- src
 - furShop
 - AboutUsPage
 - AccountPanel
 - ChairStyle1
 - ChairStyle2
 - ChairStyle3
 - ChairStyle4
 - ChairStyleTiles
 - Dashboard
 - DesignContentPanel
 - FurnitureDesignPanel
 - HomePage
 - LoginPage
 - LogoutPage
 - OrderPage
 - PreviewPanel
 - PrivacyPolicyPage
 - RegisterPage
 - SavePage
 - SettingsPage
 - Shading1
 - Shading2
 - TableStyle1

```
setExtendedState(JFrame.MAXIMIZED_BOTH);
setLayout(new BorderLayout());
JPanel mainPanel = new JPanel(new BorderLayout());
add(mainPanel, BorderLayout.CENTER);

JPanel imagePanel = paintComponent(g) {
    super.paintComponent(g);
    ImageIcon imageIcon = new ImageIcon(getClass().getResource("image/login.j
    Image image = imageIcon.getImage();
    g.drawImage(image, 0, 0, getWidth(), getHeight(), observer);
};

imagePanel.setLayout(new BorderLayout());
mainPanel.add(imagePanel, BorderLayout.NORTH);

JLabel titleLabel = new JLabel("Cozy Comfort Furniture");
titleLabel.setFont(new Font("Monotype Corsiva", Font.BOLD | Font.ITALIC, size));
titleLabel.setHorizontalTextPosition(SwingConstants.CENTER);
titleLabel.setForeground(Color.BLACK);
imagePanel.add(titleLabel, BorderLayout.NORTH);

JPanel loginPanel = createLoginPanel();
mainPanel.add(loginPanel, BorderLayout.WEST);

setSize(width: 800, height: 400);
 setLocationRelativeTo(null);
setVisible(true);
```

8:27 CRLF UTF-8 4 spaces

11:29 PM

4/9/2024

Project tree:

- .idea
- src
 - furShop
 - AboutUsPage
 - AccountPanel
 - ChairStyle1
 - ChairStyle2
 - ChairStyle3
 - ChairStyle4
 - ChairStyleTiles
 - Dashboard
 - DesignContentPanel
 - FurnitureDesignPanel
 - HomePage
 - LoginPage
 - LogoutPage
 - OrderPage
 - PreviewPanel
 - PrivacyPolicyPage
 - RegisterPage
 - SavePage
 - SettingsPage
 - Shading1
 - Shading2
 - TableStyle1

```
}
```

Usage: `lakshika`

```
private JPanel createLoginPanel() {
    JPanel panel = new JPanel(new GridBagLayout());
    panel.setBorder(BorderFactory.createLineBorder(Color.BLACK));

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridx = 0;
    gbc.gridy = 0;
    gbc.insets = new Insets(10, 10, 10, 10);

    JLabel loginHeading = new JLabel("Login");
    loginHeading.setFont(new Font("Arial", Font.BOLD, 20));
    panel.add(loginHeading, gbc);

    gbc.gridy++;
    JLabel usernameLabel = new JLabel("Username:");
    panel.add(usernameLabel, gbc);

    gbc.gridy++;
    JTextField usernameField = new JTextField(20);
    panel.add(usernameField, gbc);

    gbc.gridy++;
    gbc.gridx = 0;
    gbc.gridwidth = 2;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    JButton loginButton = new JButton("Login");
    lakshika
    loginButton.addActionListener(new ActionListener() {
        lakshika
        @Override
        public void actionPerformed(ActionEvent e) { validateLogin(); }
    });
    loginButton.setBackground(new Color(48, 17, 5));
    loginButton.setForeground(Color.WHITE);
    panel.add(loginButton, gbc);

    gbc.gridy++;
    JLabel registerLabel = new JLabel("<html><body>Don't have an account? <a href='register.html'>Join here</a></body></html>");
    registerLabel.setForeground(Color.BLUE.darker());
```

8:27 CRLF UTF-8 4 spaces

11:30 PM

4/9/2024

Project tree:

- .idea
- src
 - furShop
 - AboutUsPage
 - AccountPanel
 - ChairStyle1
 - ChairStyle2
 - ChairStyle3
 - ChairStyle4
 - ChairStyleTiles
 - Dashboard
 - DesignContentPanel
 - FurnitureDesignPanel
 - HomePage
 - LoginPage
 - LogoutPage
 - OrderPage
 - PreviewPanel
 - PrivacyPolicyPage
 - RegisterPage
 - SavePage
 - SettingsPage
 - Shading1
 - Shading2
 - TableStyle1

```
gbc.gridx = 0;
JLabel passwordLabel = new JLabel("Password:");
panel.add(passwordLabel, gbc);

gbc.gridy++;
JPasswordField passwordField = new JPasswordField(20);
panel.add(passwordField, gbc);

gbc.gridy++;
gbc.gridx = 0;
gbc.gridwidth = 2;
gbc.fill = GridBagConstraints.HORIZONTAL;
JButton loginButton = new JButton("Login");
lakshika
loginButton.addActionListener(new ActionListener() {
    lakshika
    @Override
    public void actionPerformed(ActionEvent e) { validateLogin(); }
});
loginButton.setBackground(new Color(48, 17, 5));
loginButton.setForeground(Color.WHITE);
panel.add(loginButton, gbc);

gbc.gridy++;

JLabel registerLabel = new JLabel("<html><body>Don't have an account? <a href='register.html'>Join here</a></body></html>");
```

8:27 CRLF UTF-8 4 spaces

11:30 PM

4/9/2024

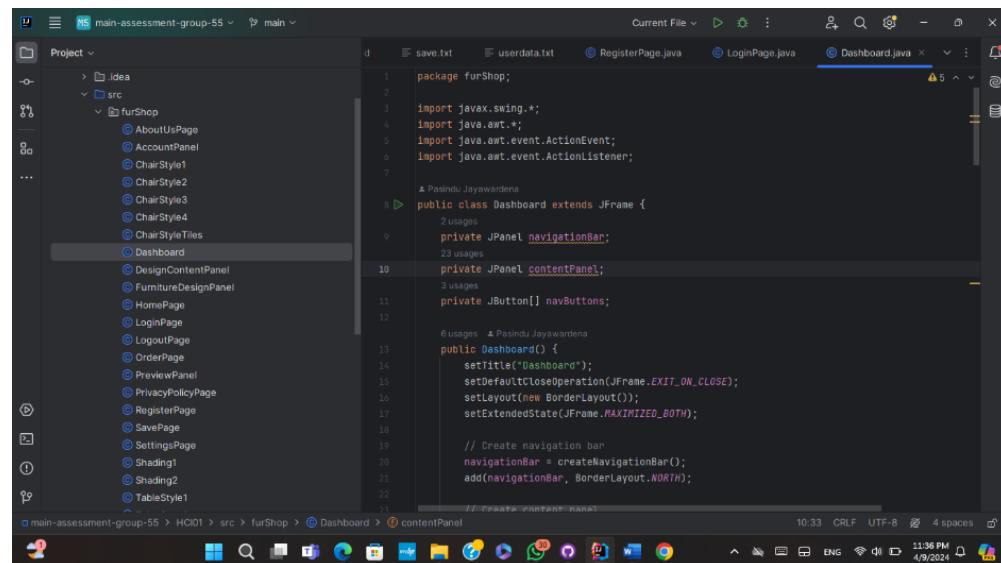
The screenshot shows a Java IDE interface with the following details:

- Project Explorer:** Shows a project structure with a .idea folder and a src folder containing packages like furShop, which contains classes such as AboutUsPage, AccountPanel, ChairStyle1, ChairStyle2, ChairStyle3, ChairStyle4, ChairStyleTiles, Dashboard, DesignContentPanel, FurnitureDesignPanel, HomePage, LoginPage, LogoutPage, OrderPage, PreviewPanel, PrivacyPolicyPage, RegisterPage, SavePage, SettingsPage, Shading1, Shading2, and TableStyle1.
- Code Editor:** The current file is LoginPage.java. The code implements a JPanel named panel. It contains a JLabel registerLabel with a specific HTML text and a mouse listener. A method validateLogin() reads from a file userDataFile using a BufferedReader and checks if the provided username and password match any entries. If found, it shows a JOptionPane message and creates a new Dashboard instance. If not found or an error occurs, it shows an invalid message or an error message respectively.
- Status Bar:** Shows the path main-assessment-group-55 > HCI01 > src > furShop > LoginPage, along with system information like 8:27, CRLF, UTF-8, 4 spaces, and a timestamp of 4/9/2024 11:30 PM.

The screenshot shows a Java IDE interface with the following details:

- Project Explorer:** Shows a project structure with a .idea folder and a src folder containing packages like furShop, which contains classes such as AboutUsPage, AccountPanel, ChairStyle1, ChairStyle2, ChairStyle3, ChairStyle4, ChairStyleTiles, Dashboard, DesignContentPanel, FurnitureDesignPanel, HomePage, LoginPage, LogoutPage, OrderPage, PreviewPanel, PrivacyPolicyPage, RegisterPage, SavePage, SettingsPage, Shading1, Shading2, and TableStyle1.
- Code Editor:** The current file is LoginPage.java. The code continues from the previous snippet, handling the successful login case by closing the reader, checking if found, and displaying a success message or an invalid user message. It then creates a new Dashboard instance. If an IOException occurs, it prints the stack trace and shows an error message. The code also includes a navigateToRegisterPage() method and a main() entry point.
- Status Bar:** Shows the path main-assessment-group-55 > HCI01 > src > furShop > LoginPage, along with system information like 8:27, CRLF, UTF-8, 4 spaces, and a timestamp of 4/9/2024 11:30 PM.

➤ Dashboard java class.



```
package furShop;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Dashboard extends JFrame {
    private JPanel navigationBar;
    private JPanel contentPanel;
    JButton[] navButtons;

    public Dashboard() {
        setTitle("Dashboard");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        setExtendedState(JFrame.MAXIMIZED_BOTH);

        // Create navigation bar
        navigationBar = createNavigationBar();
        add(navigationBar, BorderLayout.NORTH);

        // Create content panel
        contentPanel = new JPanel();
        contentPanel.setLayout(new BorderLayout());
        add(contentPanel, BorderLayout.CENTER);

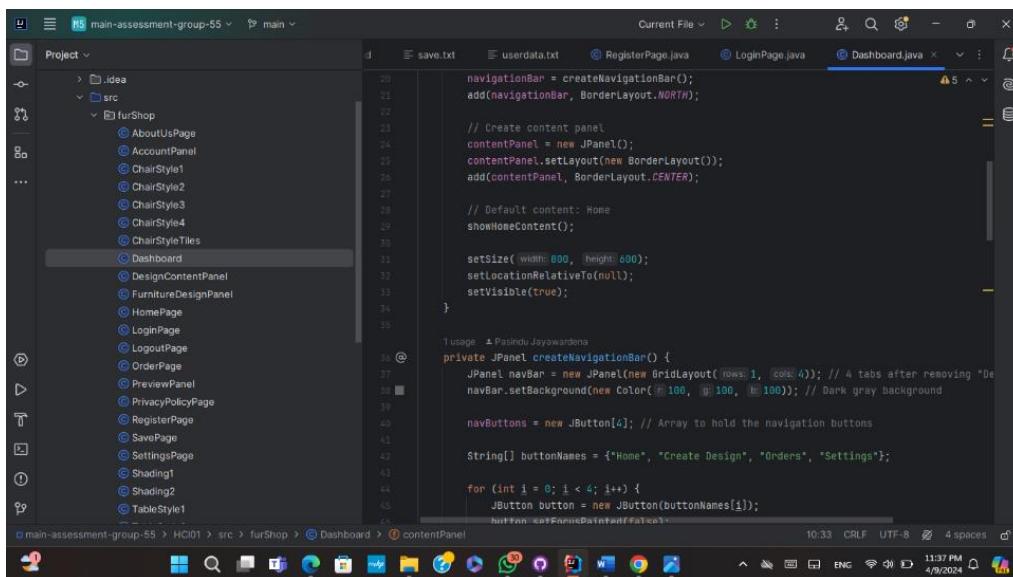
        // Default content: Home
        showHomeContent();

        setSize(800, 600);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    Usage: ▾ Pasindu Jayawardena
    private JPanel createNavigationBar() {
        JPanel navBar = new JPanel(new GridLayout(1, 4)); // 4 tabs after removing "De
        navBar.setBackground(new Color(100, 100, 100)); // Dark gray background

        navButtons = new JButton[4]; // Array to hold the navigation buttons

        String[] buttonNames = {"Home", "Create Design", "Orders", "Settings"};
        for (int i = 0; i < 4; i++) {
            JButton button = new JButton(buttonNames[i]);
            button.setOpaque(true);
        }
    }
}
```



```
navigationBar = createNavigationBar();
add(navigationBar, BorderLayout.NORTH);

// Create content panel
contentPanel = new JPanel();
contentPanel.setLayout(new BorderLayout());
add(contentPanel, BorderLayout.CENTER);

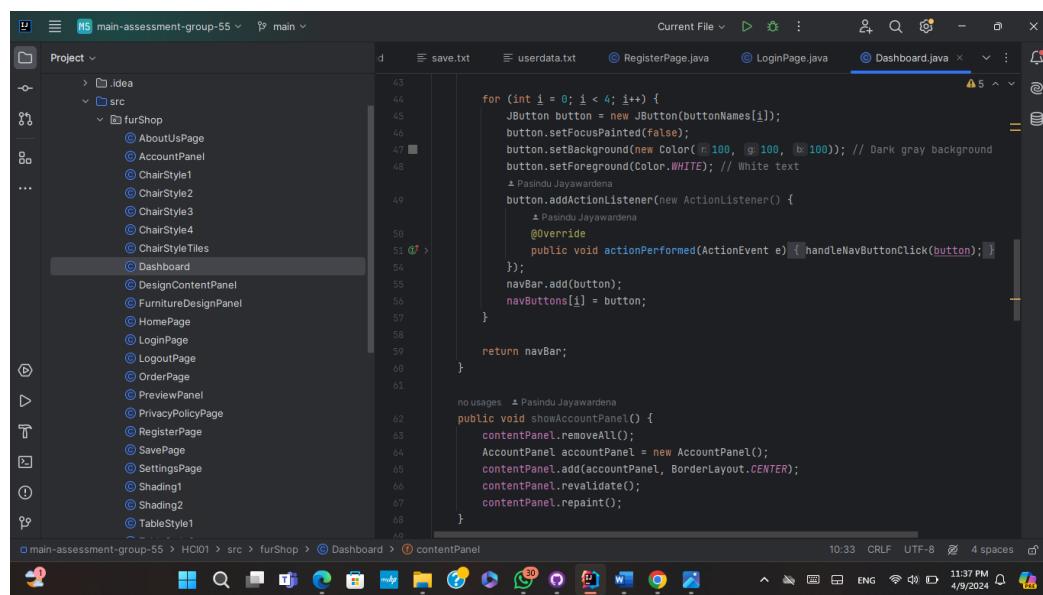
// Default content: Home
showHomeContent();

setSize(800, 600);
setLocationRelativeTo(null);
setVisible(true);

Usage: ▾ Pasindu Jayawardena
private JPanel createNavigationBar() {
    JPanel navBar = new JPanel(new GridLayout(1, 4)); // 4 tabs after removing "De
    navBar.setBackground(new Color(100, 100, 100)); // Dark gray background

    navButtons = new JButton[4]; // Array to hold the navigation buttons

    String[] buttonNames = {"Home", "Create Design", "Orders", "Settings"};
    for (int i = 0; i < 4; i++) {
        JButton button = new JButton(buttonNames[i]);
        button.setOpaque(true);
    }
}
```



```
for (int i = 0; i < 4; i++) {
    JButton button = new JButton(buttonNames[i]);
    button.setFocusPainted(false);
    button.setBackground(new Color(100, 100, 100)); // Dark gray background
    button.setForeground(Color.WHITE); // White text
}
button.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { handleNavBarClick(button); }
});
navBar.add(button);
navButtons[i] = button;
}

return navBar;
}

no usages ▾ Pasindu Jayawardena
public void showAccountPanel() {
    contentPanel.removeAll();
    AccountPanel accountPanel = new AccountPanel();
    contentPanel.add(accountPanel, BorderLayout.CENTER);
    contentPanel.revalidate();
    contentPanel.repaint();
}
```

The screenshot shows the IntelliJ IDEA interface with the code editor open to the `Dashboard.java` file. The `handleNewButtonClick` method is selected. The code handles button clicks by changing the background color of the clicked button and showing different content based on the button's text. It includes logic for "Home", "Create Design", "Orders", and "Settings".

```
private void handleNewButtonClick(JButton clickedButton) {
    for (JButton button : navButtons) {
        button.setBackground(new Color(100, 100, 100)); // Reset background color of all buttons
    }
    clickedButton.setBackground(Color.GRAY); // Highlight the clicked button

    switch (clickedButton.getText()) {
        case "Home":
            showHomeContent();
            break;
        case "Create Design":
            showCreateDesignContent();
            break;
        case "Orders":
            showOrdersPage();
            break;
        case "Settings":
            showSettingsPage();
            break;
        default:
            // Do nothing
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the `showHomeContent` method selected in the `Dashboard.java` file. The method removes all content from the panel, adds a new `HomePage`, and repaints the panel.

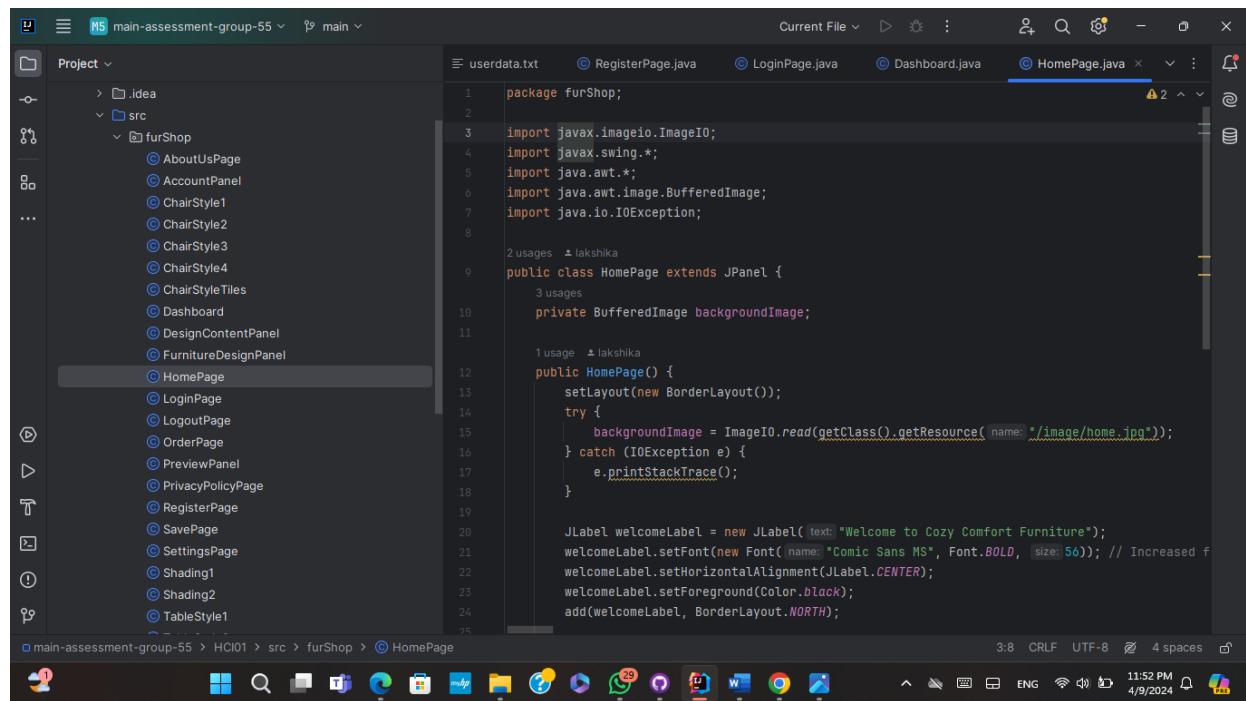
```
private void showHomeContent() {
    contentPanel.removeAll();
    HomePage homepage = new HomePage();
    contentPanel.add(homepage, BorderLayout.CENTER);
    contentPanel.revalidate();
    contentPanel.repaint();
}
```

The screenshot shows the IntelliJ IDEA interface with the `showOrdersPage` and `showSettingsPage` methods selected in the `Dashboard.java` file. Both methods follow a similar pattern of removing all content, adding a specific page, validating the panel, and repainting it.

```
private void showOrdersPage() {
    contentPanel.removeAll();
    OrderPage orderPage = new OrderPage();
    contentPanel.add(orderPage, BorderLayout.CENTER);
    contentPanel.revalidate();
    contentPanel.repaint();
}

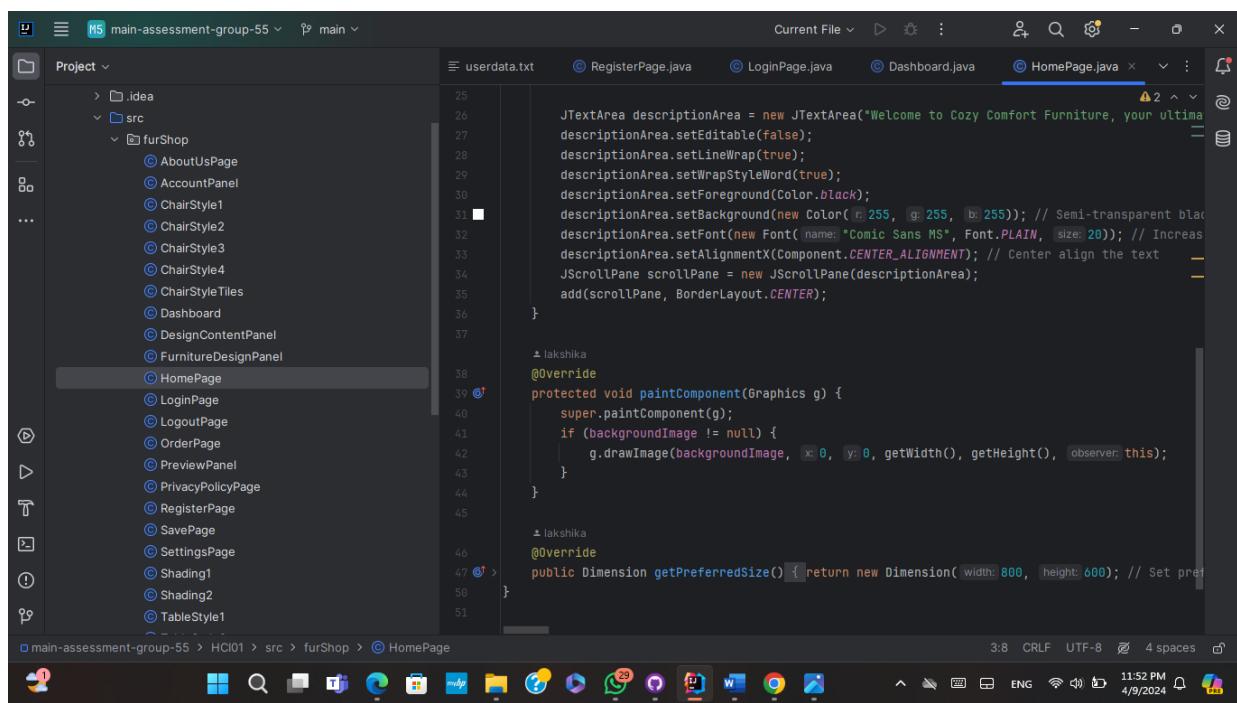
private void showSettingsPage() {
    contentPanel.removeAll();
    SettingsPage settingsPage = new SettingsPage(); // Pass Dashboard reference to SettingsPage
    contentPanel.add(settingsPage, BorderLayout.CENTER);
    contentPanel.revalidate();
    contentPanel.repaint();
}
```

➤ Home java class.



The screenshot shows the IntelliJ IDEA interface with the project 'main-assessment-group-55' open. The 'src' directory contains a package named 'furShop' which includes various Java files like AboutUsPage, AccountPanel, etc. The 'HomePage.java' file is currently selected and displayed in the editor. The code defines a class 'HomePage' that extends 'JPanel'. It sets the background image to 'home.jpg' and adds a welcome label with bold text.

```
package furShop;
import javax.imageio.ImageIO;
import javax.swing.*;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.IOException;
public class HomePage extends JPanel {
    private BufferedImage backgroundImage;
    public HomePage() {
        setLayout(new BorderLayout());
        try {
            backgroundImage = ImageIO.read(getClass().getResource("/image/home.jpg"));
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel welcomeLabel = new JLabel("Welcome to Cozy Comfort Furniture");
        welcomeLabel.setFont(new Font("Comic Sans MS", Font.BOLD, 56)); // Increased font size
        welcomeLabel.setHorizontalTextPosition(JLabel.CENTER);
        welcomeLabel.setForeground(Color.black);
        add(welcomeLabel, BorderLayout.NORTH);
    }
}
```



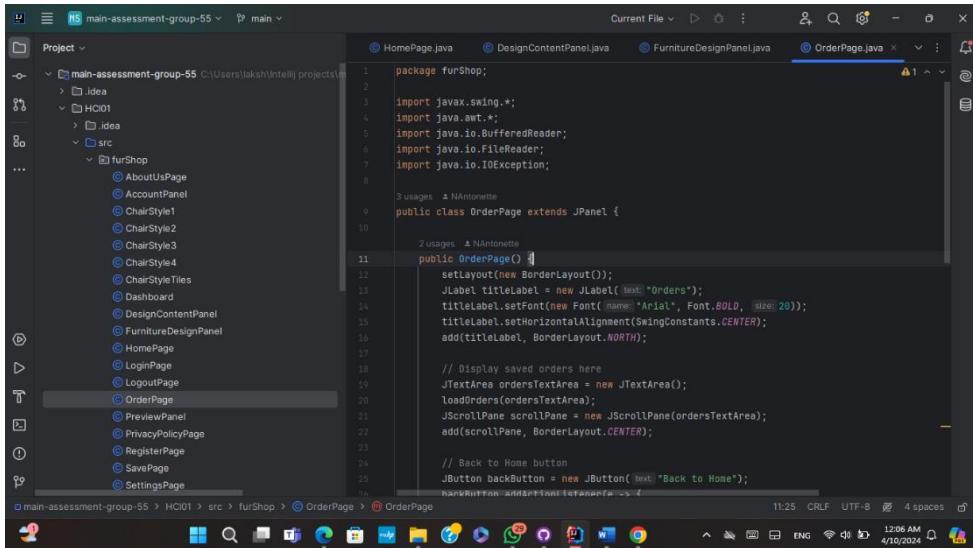
The screenshot shows the continuation of the 'HomePage.java' code from the previous screen. It includes the creation of a 'descriptionArea' JTextArea with semi-transparent black background, center alignment, and a scroll pane. It also overrides the paintComponent method to draw the background image and gets the preferred size of the panel.

```
JTextArea descriptionArea = new JTextArea("Welcome to Cozy Comfort Furniture, your ultimate destination for all your furniture needs.");
descriptionArea.setEditable(false);
descriptionArea.setLineWrap(true);
descriptionArea.setWrapStyleWord(true);
descriptionArea.setForeground(Color.black);
descriptionArea.setBackground(new Color(255, 255, 255)); // Semi-transparent black
descriptionArea.setFont(new Font("Comic Sans MS", Font.PLAIN, 20)); // Increases font size
descriptionArea.setAlignmentX(Component.CENTER_ALIGNMENT); // Center align the text
JScrollPane scrollPane = new JScrollPane(descriptionArea);
add(scrollPane, BorderLayout.CENTER);

@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (backgroundImage != null) {
        g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
    }
}

@Override
public Dimension getPreferredSize() { return new Dimension(800, 600); // Set preferred size }
```

➤ Order page java class.



Project tree:

- main-assessment-group-55
- HCI01
- src
- furShop
- AboutUsPage
- AccountPanel
- ChairStyle1
- ChairStyle2
- ChairStyle3
- ChairStyle4
- ChairStyleTiles
- Dashboard
- DesignContentPanel
- HomePage
- LoginPage
- LogoutPage
- OrderPage
- PreviewPanel
- PrivacyPolicyPage
- RegisterPage
- SavePage
- SettingsPage

Code editor (OrderPage.java):package furShop;
import javax.swing.*;
import java.awt.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

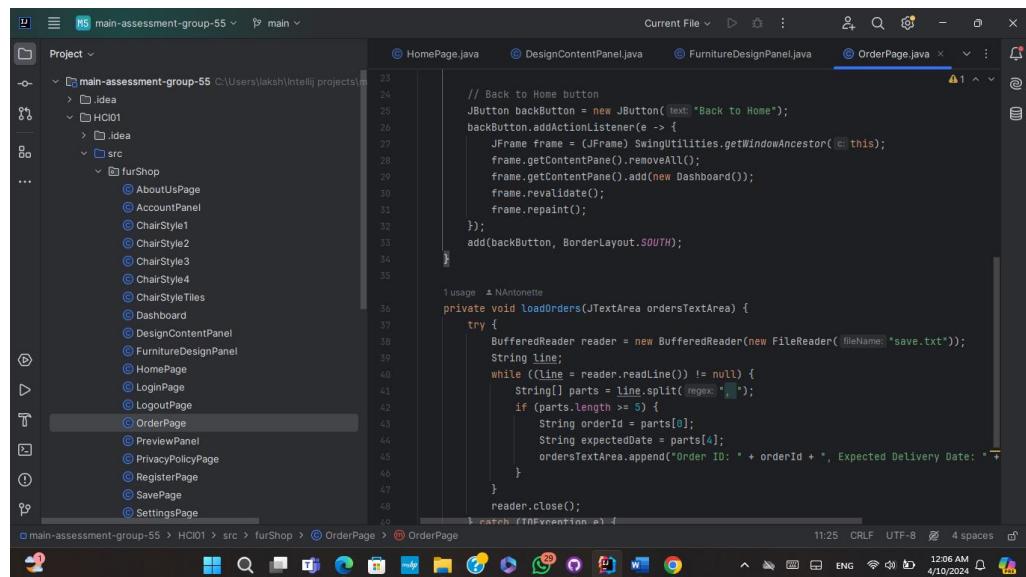
public class OrderPage extends JPanel {

 public OrderPage() {
 setLayout(new BorderLayout());
 JLabel titleLabel = new JLabel("Orders");
 titleLabel.setFont(new Font("Arial", Font.BOLD, 20));
 titleLabel.setHorizontalAlignment(SwingConstants.CENTER);
 add(titleLabel, BorderLayout.NORTH);

 // Display saved orders here
 JTextArea ordersTextArea = new JTextArea();
 loadOrders(ordersTextArea);
 JScrollPane scrollPane = new JScrollPane(ordersTextArea);
 add(scrollPane, BorderLayout.CENTER);

 // Back to Home button
 JButton backButton = new JButton("Back to Home");
 backButton.addActionListener(e -> {
 JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
 frame.getContentPane().removeAll();
 frame.getContentPane().add(new Dashboard());
 frame.revalidate();
 frame.repaint();
 });
 add(backButton, BorderLayout.SOUTH);
 }

 private void loadOrders(JTextArea ordersTextArea) {
 try {
 BufferedReader reader = new BufferedReader(new FileReader("save.txt"));
 String line;
 while ((line = reader.readLine()) != null) {
 String[] parts = line.split("\\s+");
 if (parts.length >= 5) {
 String orderId = parts[0];
 String expectedDate = parts[4];
 ordersTextArea.append("Order ID: " + orderId + ", Expected Delivery Date: " +
 expectedDate + "\n");
 }
 }
 reader.close();
 } catch (IOException e) {
 e.printStackTrace();
 }
 }
}

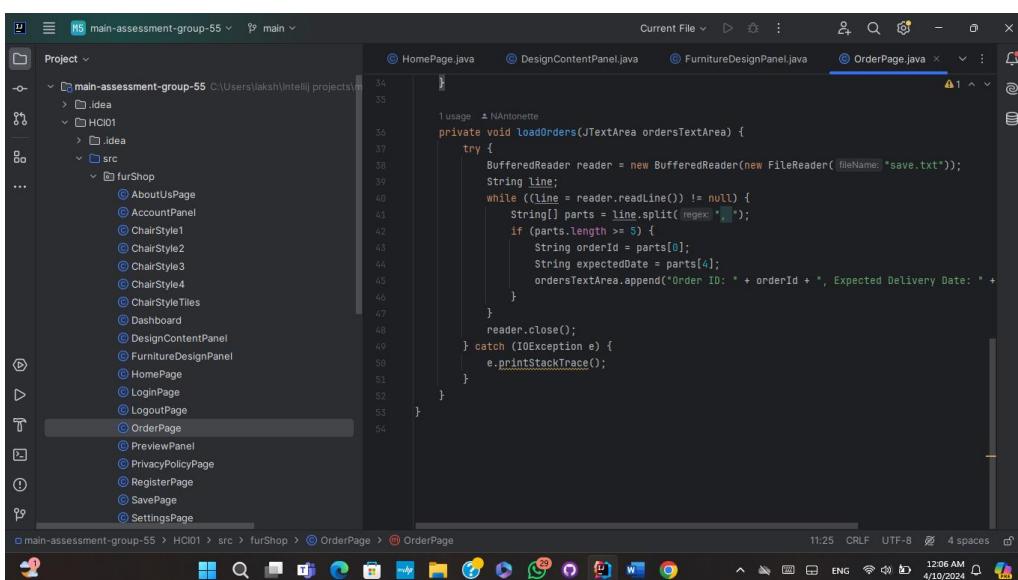


Project tree:

- main-assessment-group-55
- HCI01
- src
- furShop
- AboutUsPage
- AccountPanel
- ChairStyle1
- ChairStyle2
- ChairStyle3
- ChairStyle4
- ChairStyleTiles
- Dashboard
- DesignContentPanel
- FurnitureDesignPanel
- HomePage
- LoginPage
- LogoutPage
- OrderPage
- PreviewPanel
- PrivacyPolicyPage
- RegisterPage
- SavePage
- SettingsPage

Code editor (OrderPage.java):// Back to Home button
JButton backButton = new JButton("Back to Home");
backButton.addActionListener(e -> {
 JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
 frame.getContentPane().removeAll();
 frame.getContentPane().add(new Dashboard());
 frame.revalidate();
 frame.repaint();
});
add(backButton, BorderLayout.SOUTH);

private void loadOrders(JTextArea ordersTextArea) {
 try {
 BufferedReader reader = new BufferedReader(new FileReader("save.txt"));
 String line;
 while ((line = reader.readLine()) != null) {
 String[] parts = line.split("\\s+");
 if (parts.length >= 5) {
 String orderId = parts[0];
 String expectedDate = parts[4];
 ordersTextArea.append("Order ID: " + orderId + ", Expected Delivery Date: " +
 expectedDate + "\n");
 }
 }
 reader.close();
 } catch (IOException e) {
 e.printStackTrace();
 }
}



Project tree:

- main-assessment-group-55
- HCI01
- src
- furShop
- AboutUsPage
- AccountPanel
- ChairStyle1
- ChairStyle2
- ChairStyle3
- ChairStyle4
- ChairStyleTiles
- Dashboard
- DesignContentPanel
- FurnitureDesignPanel
- HomePage
- LoginPage
- LogoutPage
- OrderPage
- PreviewPanel
- PrivacyPolicyPage
- RegisterPage
- SavePage
- SettingsPage

Code editor (OrderPage.java):1 usage ▲ NAntonette
private void loadOrders(JTextArea ordersTextArea) {
 try {
 BufferedReader reader = new BufferedReader(new FileReader("save.txt"));
 String line;
 while ((line = reader.readLine()) != null) {
 String[] parts = line.split("\\s+");
 if (parts.length >= 5) {
 String orderId = parts[0];
 String expectedDate = parts[4];
 ordersTextArea.append("Order ID: " + orderId + ", Expected Delivery Date: " +
 expectedDate + "\n");
 }
 }
 reader.close();
 } catch (IOException e) {
 e.printStackTrace();
 }
}

➤ Setting page java class.

The screenshot shows the Java code for the `SettingsPage` class. The code initializes a `JPanel` with a `BorderLayout`, sets its background color to white, and adds a title label and several buttons for account, privacy policy, terms and conditions, and about us.

```
package furShop;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SettingsPage extends JPanel {
    public SettingsPage() {
        setLayout(new BorderLayout());
        setBackground(Color.WHITE); // Set background color

        // Create the settings panel
        JPanel settingsPanel = new JPanel();
        settingsPanel.setLayout(new GridLayout(5, 1));
        settingsPanel.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));
        settingsPanel.setBackground(Color.WHITE); // Set background color

        // Add the settings options
        JLabel titleLabel = new JLabel("Settings");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setForeground(Color.DARK_GRAY); // Set text color
        settingsPanel.add(titleLabel);

        JButton accountButton = createStyledButton("Account");
        JButton privacyPolicyButton = createStyledButton("Privacy Policy");
        JButton termsButton = createStyledButton("Terms and Conditions");
        JButton aboutUsButton = createStyledButton("About Us");

        accountButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToAccount(); }
        });
        privacyPolicyButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToPrivacyPolicy(); }
        });
        termsButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToTermsConditions(); }
        });
        aboutUsButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToAboutUs(); }
        });

        // Add the settings panel to the center of this panel
        add(settingsPanel, BorderLayout.CENTER);

        // Add a logout button at the bottom
        JButton logoutButton = new JButton("Logout");
        add(logoutButton, BorderLayout.SOUTH);
    }

    private JButton createStyledButton(String text) {
        JButton button = new JButton(text);
        button.setFont(new Font("Arial", Font.PLAIN, 16));
        button.setForeground(Color.DARK_GRAY);
        button.setBackground(Color.WHITE);
        return button;
    }

    private void navigateToAccount() {
        // Navigate to the Account page
    }

    private void navigateToPrivacyPolicy() {
        // Navigate to the Privacy Policy page
    }

    private void navigateToTermsConditions() {
        // Navigate to the Terms and Conditions page
    }

    private void navigateToAboutUs() {
        // Navigate to the About Us page
    }
}
```

The screenshot shows the continuation of the `SettingsPage` class. It includes the implementation of the `actionPerformed` method for each button, which calls the corresponding navigation methods.

```
    JButton accountButton = createStyledButton("Account");
    JButton privacyPolicyButton = createStyledButton("Privacy Policy");
    JButton termsButton = createStyledButton("Terms and Conditions");
    JButton aboutUsButton = createStyledButton("About Us");

    accountButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) { navigateToAccount(); }
    });
    privacyPolicyButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) { navigateToPrivacyPolicy(); }
    });
    termsButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) { navigateToTermsConditions(); }
    });
    aboutUsButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) { navigateToAboutUs(); }
    });

    // Add the settings panel to the center of this panel
    add(settingsPanel, BorderLayout.CENTER);

    // Add a logout button at the bottom
    JButton logoutButton = new JButton("Logout");
    add(logoutButton, BorderLayout.SOUTH);
}
```

The screenshot shows the final version of the `SettingsPage` class. The code is identical to the previous screenshots but includes additional annotations and comments to explain the logic.

```
package furShop;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SettingsPage extends JPanel {
    public SettingsPage() {
        setLayout(new BorderLayout());
        setBackground(Color.WHITE); // Set background color

        // Create the settings panel
        JPanel settingsPanel = new JPanel();
        settingsPanel.setLayout(new GridLayout(5, 1));
        settingsPanel.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));
        settingsPanel.setBackground(Color.WHITE); // Set background color

        // Add the settings options
        JLabel titleLabel = new JLabel("Settings");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setForeground(Color.DARK_GRAY); // Set text color
        settingsPanel.add(titleLabel);

        JButton accountButton = createStyledButton("Account");
        JButton privacyPolicyButton = createStyledButton("Privacy Policy");
        JButton termsButton = createStyledButton("Terms and Conditions");
        JButton aboutUsButton = createStyledButton("About Us");

        accountButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToAccount(); }
        });
        privacyPolicyButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToPrivacyPolicy(); }
        });
        termsButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToTermsConditions(); }
        });
        aboutUsButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { navigateToAboutUs(); }
        });

        // Add the settings panel to the center of this panel
        add(settingsPanel, BorderLayout.CENTER);

        // Add a logout button at the bottom
        JButton logoutButton = new JButton("Logout");
        add(logoutButton, BorderLayout.SOUTH);
    }

    private JButton createStyledButton(String text) {
        JButton button = new JButton(text);
        button.setFont(new Font("Arial", Font.PLAIN, 16));
        button.setForeground(Color.DARK_GRAY);
        button.setBackground(Color.WHITE);
        return button;
    }

    private void navigateToAccount() {
        // Navigate to the Account page
    }

    private void navigateToPrivacyPolicy() {
        // Navigate to the Privacy Policy page
    }

    private void navigateToTermsConditions() {
        // Navigate to the Terms and Conditions page
    }

    private void navigateToAboutUs() {
        // Navigate to the About Us page
    }
}
```

The screenshot shows a Java IDE interface with the following details:

- Project:** main-assessment-group-55
- Current File:** SettingsPage.java
- Code Content:** The code defines a class named `SettingsPage` which extends `DesignContentPanel`. It contains methods for adding a settings panel and a logout button, and a navigation method `navigateToLogout()`.
- Toolbars and Status Bar:** The status bar at the bottom shows the current file is `SettingsPage.java`, the time is 6:38 AM, and the date is 4/10/2024.

The screenshot shows a Java IDE interface with the following details:

- Project:** main-assessment-group-55
- Current File:** SettingsPage.java
- Code Content:** The code defines a class named `SettingsPage` which extends `DesignContentPanel`. It includes a navigation method `navigateToDashboard()` and a helper method `createStyledButton` for creating styled buttons.
- Toolbars and Status Bar:** The status bar at the bottom shows the current file is `SettingsPage.java`, the time is 6:38 AM, and the date is 4/10/2024.

The screenshot shows a Java IDE interface with the following details:

- Project:** main-assessment-group-55
- Current File:** SettingsPage.java
- Code Content:** The code defines a class named `SettingsPage` which extends `DesignContentPanel`. It includes methods for navigating to account, privacy policy, and terms conditions.
- Toolbars and Status Bar:** The status bar at the bottom shows the current file is `SettingsPage.java`, the time is 6:38 AM, and the date is 4/10/2024.

```
DesignContentPanel.java FurnitureDesignPanel.java OrderPage.java SettingsPage.java

AboutUsPage 135
AccountPanel 136
ChairStyle1 137
ChairStyle2 138
ChairStyle3 139
ChairStyle4 140
ChairStyleTiles 141
Dashboard 142
DesignContentPanel 143
FurnitureDesignPanel 144
HomePage 145
LoginPage 146
LogoutPage 147
OrderPage 148
PreviewPanel 149
PrivacyPolicyPage 150
RegisterPage 151
SavePage 152
SettingsPage 153
Shading1 154
Shading2 155
TableStyle1 156
TableStyle2 157
TableStyle3 158
TableStyle4 159

private void navigateToAboutUs() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    frame.getContentPane().add(new AboutUsPage());
    frame.revalidate();
    frame.repaint();
}

private void navigateToLogout() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    frame.getContentPane().add(new LogoutPage(frame));
    frame.revalidate();
    frame.repaint();
}

private void navigateToDashboard() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    Dashboard dashboard = new Dashboard();
    frame.getContentPane().add(dashboard, BorderLayout.CENTER);
    frame.revalidate();
    frame.repaint();
}

main-assessment-group-55 > HCI01 > src > furShop > SettingsPage > navigateToAboutUs
```

➤ chair style titles page java class.

```
package furShop;
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class ChairStyleTiles extends JPanel {
    public ChairStyleTiles() {
        setLayout(new BorderLayout()); // Use BorderLayout to position components

        // Add back button at the top
        JButton backButton = new JButton("Back");
        backButton.addActionListener(e -> navigateToDashboard());
        add(backButton, BorderLayout.NORTH);

        JPanel tilesPanel = new JPanel();
        tilesPanel.setLayout(new GridLayout(2, 4, 10, 10)); // 2 rows, 4 columns

        // Create tiles for chair styles
        JPanel chairStyle1Panel = createTilePanel("Chair Style 1", "This is the first style");
        JPanel chairStyle2Panel = createTilePanel("Chair Style 2", "Explore our second style");
        JPanel chairStyle3Panel = createTilePanel("Chair Style 3", "This round style is popular");
        JPanel chairStyle4Panel = createTilePanel("Chair Style 4", "This anti-style is unique");
        JPanel chairStyle1Panel = createTilePanel("Table Style 1", "This elegant table complements our chairs");

        tilesPanel.add(chairStyle1Panel);
        tilesPanel.add(chairStyle2Panel);
        tilesPanel.add(chairStyle3Panel);
        tilesPanel.add(chairStyle4Panel);
        tilesPanel.add(chairStyle1Panel);
    }

    private JPanel createTilePanel(String styleName, String description) {
        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout());
        panel.setBorder(new EmptyBorder(10, 10, 10, 10));
        panel.add(new JLabel(styleName), BorderLayout.NORTH);
        panel.add(new JLabel(description), BorderLayout.CENTER);
        return panel;
    }

    void navigateToDashboard() {
        // Implementation for navigating to the dashboard
    }
}
```

```
17 JPanel tilesPanel = new JPanel();
18 tilesPanel.setLayout(new GridLayout(2, 4, 10, 10)); // 2 rows, 4 columns
19
20 // Create tiles for chair styles
21 JPanel chairStyle1Panel = createTilePanel("Chair Style 1", "This is the new", JButton.class);
22 JPanel chairStyle2Panel = createTilePanel("Chair Style 2", "Explore our col", JButton.class);
23 JPanel chairStyle3Panel = createTilePanel("Chair Style 3", "This rounded ch", JButton.class);
24 JPanel chairStyle4Panel = createTilePanel("Chair Style 4", "This antique-st", JButton.class);
25 JPanel tableStyle1Panel = createTilePanel("Table Style 1", "This elegant ar", JButton.class);
26 JPanel tableStyle2Panel = createTilePanel("Table Style 2", "This table fea", JButton.class);
27 JPanel tableStyle3Panel = createTilePanel("Table Style 3", "This table is s", JButton.class);
28 JPanel tableStyle4Panel = createTilePanel("Table Style 4", "This versatile", JButton.class);
29
30 tilesPanel.add(chairStyle1Panel);
31 tilesPanel.add(chairStyle2Panel);
32 tilesPanel.add(chairStyle3Panel);
33 tilesPanel.add(chairStyle4Panel);
34 tilesPanel.add(tableStyle1Panel);
35 tilesPanel.add(tableStyle2Panel);
36 tilesPanel.add(tableStyle3Panel);
37 tilesPanel.add(tableStyle4Panel);
38
39 add(tilesPanel, BorderLayout.CENTER);
```

```
41
42
43 @
44 JPanel tilePanel = new JPanel();
45 tilePanel.setLayout(new BorderLayout()); // Use BorderLayout to stack components vertically
46 tilePanel.setPreferredSize(new Dimension(200, 200)); // Adjust tile size as needed
47 tilePanel.setBorder(BorderFactory.createLineBorder(Color.BLACK));
48 tilePanel.setBackground(Color.WHITE); // Set background color
49 tilePanel.setToolTipText(description); // Add tooltip with description
50
51 JLabel nameLabel = new JLabel(styleName, SwingConstants.CENTER);
52 nameLabel.setFont(new Font("Arial", Font.BOLD, 16)); // Adjust font size and style
53 JLabel descLabel = new JLabel(<html>" + description.replaceAll("\\n", "<br>"), replacement: "<br>");
54 descLabel.setFont(new Font("Arial", Font.PLAIN, 14)); // Adjust font size and style
55
56 tilePanel.add(nameLabel, BorderLayout.NORTH);
57 tilePanel.add(descLabel, BorderLayout.CENTER);
58
59 // Add hover effect
60
61 tilePanel.addMouseListener(new MouseAdapter() {
62     @Override
63     public void mouseEntered(MouseEvent e) {
64         tilePanel.setBackground(new Color(220, 220, 220)); // Light gray background or
65     }
66
67     @Override
68     public void mouseExited(MouseEvent e) {
69         tilePanel.setBackground(Color.WHITE); // Restore default background color
70     }
71
72     @Override
73     public void mouseClicked(MouseEvent e) { navigateTo(component); }
74 };
75
76 return tilePanel;
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

```
65
66     @diminawodani
67     @Override
68     public void mouseExited(MouseEvent e) {
69         tilePanel.setBackground(Color.WHITE); // Restore default background color
70     }
71
72     @diminawodani
73     @Override
74     public void mouseClicked(MouseEvent e) { navigateTo(component); }
75
76     return tilePanel;
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

The screenshot shows the IntelliJ IDEA interface with the 'main' module selected. The left sidebar displays the project structure under 'Project'. The current file, 'ChairStyleTiles.java', is open in the editor. The code implements a JPanel that adds a Dashboard component to a JFrame. It includes a main method that creates a frame, sets its title, and adds a ChairStyleTiles panel to it. The code uses SwingUtilities.invokeLater to ensure the UI is updated correctly.

```
sage ▲ dimnawodani
private void navigateToDashboard() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    // Assuming Dashboard class is available, replace it with the actual class name if it's different
    Dashboard dashboard = new Dashboard();
    frame.getContentPane().add(dashboard, BorderLayout.CENTER); // Add dashboard to center
    frame.revalidate();
    frame.repaint();
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        JFrame frame = new JFrame("Chair Styles");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setExtendedState(JFrame.MAXIMIZED_BOTH);
        ChairStyleTiles chairStyleTiles = new ChairStyleTiles();
        frame.add(chairStyleTiles);
        frame.setVisible(true);
    });
}
```

➤ Chair style 1 java class.

The screenshot shows the IntelliJ IDEA interface with the 'main' module selected. The left sidebar displays the project structure under 'Project'. The current file, 'ChairStyle1.java', is open in the editor. This class extends JPanel and implements GLEventListener, using JOGL libraries to handle OpenGL rendering. It initializes GLU glu and defines variables for angleX, angleY, scaleFactor, and partColors. The code includes imports for various Java and JOGL packages.

```
package furShop;

import com.jogamp.opengl.*;
import com.jogamp.opengl.awt.GLJPanel;
import com.jogamp.opengl.glu.GLU;
import com.jogamp.opengl.util.FPSAnimator;

import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

3 usages ▲ NAntonette
public class ChairStyle1 extends JPanel implements GLEventListener {
    2 usages
    private GLU glu;
    2 usages
    private float angleX = 0;
    2 usages
    private float angleY = 0;
    4 usages
    private float scaleFactor = 1.0f;
    8 usages
    private Color[] partColors = new Color[6];
    4 usages
    private Color roomColor = Color.WHITE;
```

```
private Color[] partColors = new Color[6];
private Color roomColor = Color.WHITE;
private GLPanel panel;
private boolean rotate = true; // Flag to control rotation

public ChairStyle1() {
    setPreferredSize(new Dimension( width: 1000, height: 600)); // Adjust panel size

    for (int i = 0; i < partColors.length; i++) {
        partColors[i] = Color.GRAY;
    }

    GLProfile profile = GLProfile.getDefault();
    GLCapabilities capabilities = new GLCapabilities(profile);
    panel = new GLPanel(capabilities);
    panel.addGLEventListener(this);
    panel.setBackground(roomColor);

    FPSAnimator animator = new FPSAnimator(panel, 60);
    animator.start();

    setLayout(new BorderLayout());
    add(panel, BorderLayout.CENTER);
}
```

```
animator.start();

setLayout(new BorderLayout());
add(panel, BorderLayout.CENTER);

glu = new GLU();

JPanel bottomPanel = new JPanel(new BorderLayout());

JButton rotateButton = new JButton("View 2D"); // Change label
rotateButton.setPreferredSize(new Dimension( width: 150, height: 40)); // Adjust button size
rotateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        rotate = !rotate; // Toggle rotation flag
        rotateButton.setText(rotate ? "View 2D" : "View 3D"); // Update button text
    }
});
bottomPanel.add(rotateButton, BorderLayout.WEST);

// Create a dropdown menu for selecting colors for each part
JComboBox<String> partDropdown = new JComboBox<>(new String[]{"Seat Color", "Backrest", "Legs", "Cushion", "Pillows", "Frame", "Hardware"});
partDropdown.setPreferredSize(new Dimension( width: 150, height: 30)); // Adjust dropdown size
partDropdown.addActionListener(new ActionListener() {
```

```
@Override
public void actionPerformed(ActionEvent e) {
    JComboBox<String> comboBox = (JComboBox<String>) e.getSource();
    int selectedIndex = comboBox.getSelectedIndex();
    Color selectedColor = JColorChooser.showDialog( component: null, title: "Choose Color" );
    if (selectedColor != null) {
        partColors[selectedIndex] = selectedColor;
        panel.repaint(); // Repaint the GLPanel to reflect color changes
    }
}
bottomPanel.add(partDropdown, BorderLayout.CENTER);

JSlider scaleSlider = new JSlider(JSlider.HORIZONTAL, min: 0, max: 200, value: 100);
scaleSlider.setPreferredSize(new Dimension( width: 300, height: 40)); // Adjust slider size
scaleSlider.setMajorTickSpacing(10);
scaleSlider.setPaintTicks(true);
scaleSlider.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        JSlider source = (JSlider) e.getSource();
        if (!source.getValueIsAdjusting()) {
            int value = source.getValue();
            scaleFactor = value / 100;
        }
    }
});
```

The screenshot shows the IntelliJ IDEA interface with the project 'main-assessment-group-55' open. The 'src' directory contains a 'furShop' package with various Java files. The 'ChairStyle1.java' file is currently selected and displayed in the editor. The code implements a functionality for adjusting chair styles, including saving designs and navigating between different styles.

```
if (!source.getValueIsAdjusting()) {
    int value = source.getValue();
    scaleFactor = value / 100f;
}

JPanel adjustSizePanel = new JPanel();
JLabel adjustSizeLabel = new JLabel("Adjust size:");
adjustSizePanel.add(adjustSizeLabel);
adjustSizePanel.add(scaleSlider);
bottomPanel.add(adjustSizePanel, BorderLayout.EAST);

// Add Save button
 JButton saveButton = new JButton("Save Design");
saveButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { showSavePage(); }
});
bottomPanel.add(saveButton, BorderLayout.SOUTH);

// Add Back button
 JButton backButton = new JButton("Back");
backButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { showBackPage(); }
});
bottomPanel.add(backButton, BorderLayout.NORTH);
```

The screenshot continues to show the 'ChairStyle1.java' code. It includes logic for navigating between different parts of the application, such as 'ChairStyleTiles', 'SavePage', and 'Shading'. The code uses annotations like @Override and implements ActionListener for various buttons.

```
@Override
public void actionPerformed(ActionEvent e) { navigateToChairStyleTiles(); }
};

bottomPanel.add(backButton, BorderLayout.NORTH);

// Add Next button
 JButton nextButton = new JButton("Next");
nextButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { navigateToSavePage(); }
});
bottomPanel.add(nextButton, BorderLayout.SOUTH);

// Add Shade button
 JButton shadingButton = new JButton("Shade");
shadingButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { navigateToShading1(); }
});

// Adding Shade button to the top right corner
 JPanel topRightPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
topRightPanel.add(shadingButton);
add(topRightPanel, BorderLayout.NORTH);
```

The screenshot continues to show the 'ChairStyle1.java' code. It includes logic for navigating between different parts of the application, such as 'ChairStyleTiles', 'SavePage', and 'Shading'. The code uses annotations like @Override and implements ActionListener for various buttons.

```
// Adding Shade button to the top right corner
 JPanel topRightPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
topRightPanel.add(shadingButton);
add(topRightPanel, BorderLayout.NORTH);

add(bottomPanel, BorderLayout.SOUTH);

}

1 usage ▾ NAntonette
private void navigateToChairStyleTiles() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    ChairStyleTiles chairStyleTiles = new ChairStyleTiles();
    frame.getContentPane().add(chairStyleTiles);
    frame.revalidate();
    frame.repaint();
}

2 usages ▾ NAntonette
private void navigateToSavePage() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    SavePage savePage = new SavePage();
    frame.getContentPane().add(savePage);
    frame.revalidate();
    frame.repaint();
}
```

```
private void showSavePage() { navigateToSavePage(); }

private void navigateToShading1() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
    frame.getContentPane().removeAll();
    Shading1 shading1 = new Shading1();
    frame.getContentPane().add(shading1);
    frame.revalidate();
    frame.repaint();
}

@Override
public void init(GLAutoDrawable drawable) {
    GL2 gl = drawable.getGL().getGL2();
    gl.glClearColor(red: roomColor.getRed() / 255f, green: roomColor.getGreen() / 255f,
                    blue: roomColor.getBlue() / 255f, alpha: 1.0f); // Set background color
    gl.glEnable(GL2.GL_DEPTH_TEST);
}

@Override
public void display(GLAutoDrawable drawable) {
    GL2 gl = drawable.getGL().getGL2();
    gl.glClear(mask: GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);
}
```

```
gl.glClear(mask: GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);
gl.glLoadIdentity();
gl.glTranslatef(x: 0.0f, y: 0.0f, z: -5.0f);
if (rotate) { // Check if rotation is enabled
    gl.glRotatef(angleX, x: 1.0f, y: 0.0f, z: 0.0f);
    gl.glRotatef(angleY, x: 0.0f, y: 1.0f, z: 0.0f);
    angleX += 0.5f;
    angleY += 0.5f;
}
gl.glScalef(scaleFactor, scaleFactor, scaleFactor);
drawChair(gl);

@Override
public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
    GL2 gl = drawable.getGL().getGL2();
    gl.glViewport(x: 0, y: 0, width, height);
    gl.glMatrixMode(GL2.GL_PROJECTION);
    gl.glLoadIdentity();
    double aspectRatio = (double) width / (double) height;
    glu.gluPerspective(fovy: 45, aspectRatio, zNear: 1, zFar: 100);
    gl.glMatrixMode(GL2.GL_MODELVIEW);
}

@Override
public void dispose(GLAutoDrawable drawable) {
    // Cleanup resources...
}
```

```
private void drawChair(GL2 gl) {
    gl.glColor3fv(partColors[0].getRGBComponents(compArray: null), v_offset: 0);
    gl.glPushMatrix();
    gl.glTranslatef(x: 0.0f, y: -1.0f, z: 0.0f);
    gl.glScalef(x: 1.5f, y: 0.2f, z: 1.5f);
    drawCube(gl);
    gl.glPopMatrix();

    gl.glColor3fv(partColors[1].getRGBComponents(compArray: null), v_offset: 0);
    gl.glPushMatrix();
    gl.glTranslatef(x: 0.0f, y: 0.5f, z: -0.75f);
    gl.glScalef(x: 1.5f, y: 1.0f, z: 0.2f);
    drawCube(gl);
    gl.glPopMatrix();

    for (int i = 2; i < partColors.length; i++) {
        gl.glColor3fv(partColors[i].getRGBComponents(compArray: null), v_offset: 0);
        gl.glPushMatrix();
        float legX = (i % 2 == 0) ? -0.75f : 0.75f;
        float legZ = (i < 6) ? -0.75f : 0.75f;
        drawLeg(gl, legX, legZ);
        gl.glPopMatrix();
    }
}
```

```
gl.glColor3fv(partColors[i].getRGBComponents( compArray: null ), v_offset: 0 ) 13 5
gl.glPushMatrix();
float legX = (i % 2 == 0) ? -0.75f : 0.75f;
float legZ = (i < 4) ? -0.75f : 0.75f;
gl.glTranslate(legX, y: -1.5f, legZ);
drawLeg(gl);
gl.glPopMatrix();
}
}

1 usage ± NAntonette
private void drawLeg(GL2 gl) {
    gl.glPushMatrix();
    gl.glScalef( x: 0.2f, y: 1.5f, z: 0.2f );
    drawCube(gl);
    gl.glPopMatrix();
}

3 usages ± NAntonette
private void drawCube(GL2 gl) {
    gl.glBegin(GL2.GL_QUADS);
    gl.glVertex3f( x: -0.5f, y: -0.5f, z: 0.5f );
    gl.glVertex3f( x: 0.5f, y: -0.5f, z: 0.5f );
    gl.glVertex3f( x: 0.5f, y: 0.5f, z: 0.5f );
    gl.glVertex3f( x: -0.5f, y: 0.5f, z: 0.5f );

    gl.glVertex3f( x: -0.5f, y: -0.5f, z: -0.5f );
    gl.glVertex3f( x: 0.5f, y: -0.5f, z: -0.5f );
    gl.glVertex3f( x: 0.5f, y: 0.5f, z: -0.5f );
    gl.glVertex3f( x: -0.5f, y: 0.5f, z: -0.5f );
}

gl.glEnd();
```

```
gl.glVertex3f( x: -0.5f, y: -0.5f, z: -0.5f );
gl.glVertex3f( x: 0.5f, y: -0.5f, z: -0.5f );
gl.glVertex3f( x: 0.5f, y: 0.5f, z: -0.5f );
gl.glVertex3f( x: -0.5f, y: 0.5f, z: -0.5f );

gl.glVertex3f( x: -0.5f, y: 0.5f, z: 0.5f );
gl.glVertex3f( x: 0.5f, y: 0.5f, z: 0.5f );
gl.glVertex3f( x: 0.5f, y: -0.5f, z: 0.5f );
gl.glVertex3f( x: -0.5f, y: -0.5f, z: 0.5f );

gl.glVertex3f( x: 0.5f, y: -0.5f, z: -0.5f );
gl.glVertex3f( x: 0.5f, y: 0.5f, z: -0.5f );
gl.glVertex3f( x: -0.5f, y: 0.5f, z: -0.5f );
gl.glVertex3f( x: -0.5f, y: -0.5f, z: -0.5f );

gl.glVertex3f( x: -0.5f, y: -0.5f, z: 0.5f );
gl.glVertex3f( x: 0.5f, y: -0.5f, z: 0.5f );
gl.glVertex3f( x: 0.5f, y: 0.5f, z: 0.5f );
gl.glVertex3f( x: -0.5f, y: 0.5f, z: 0.5f );
gl.glEnd();
```

➤ Table style 1 java class.

```

package furshop;
import com.jogamp.opengl.*;
import com.jogamp.opengl.awt.GLJPanel;
import com.jogamp.opengl.glu.GLU;
import com.jogamp.opengl.util.FPSAnimator;

import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

jayawardhanadskp
public class TableStyle1 extends JPanel implements GLEventListener {
    2 usages
    private GLU glu;
    2 usages
    private float angleX = 0;
    2 usages
    private float angleY = 0;
    4 usages
    private float scaleFactor = 1.0f;
    7 usages
    private Color[] partColors = new Color[6];
    4 usages
    private Color roomColor = Color.WHITE;

```

```

private Color[] partColors = new Color[6];
private Color roomColor = Color.WHITE;
private GLPanel panel;
4 usages
private boolean rotate = true; // Flag to control rotation

3 usages jayawardhanadskp
public Tablestyle1() {
    setPreferredSize(new Dimension( width: 1000, height: 600)); // Adjust panel size

    for (int i = 0; i < partColors.length; i++) {
        partColors[i] = Color.GRAY;
    }

    GLProfile profile = GLProfile.getDefault();
    GLCapabilities capabilities = new GLCapabilities(profile);
    panel = new GLJPanel(capabilities);
    panel.addGLEventListener(this);
    panel.setBackground(roomColor);

    FPSAnimator animator = new FPSAnimator(panel, [fps: 60]);
    animator.start();

    setLayout(new BorderLayout());
    add(panel, BorderLayout.CENTER);
}

```

```

FPSAnimator animator = new FPSAnimator(panel, [fps: 60]);
animator.start();

setLayout(new BorderLayout());
add(panel, BorderLayout.CENTER);

glu = new GLU();

JPanel bottomPanel = new JPanel(new BorderLayout());

JButton rotateButton = new JButton( text: "View 2D"); // Change label
rotateButton.setPreferredSize(new Dimension( width: 150, height: 40)); // Adjust button size
jayawardhanadskp
rotateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        rotate = !rotate; // Toggle rotation flag
        rotateButton.setText(rotate ? "View 2D" : "View 3D"); // Update button text
    }
});
bottomPanel.add(rotateButton, BorderLayout.WEST);

// Create a dropdown menu for selecting colors for each part
JComboBox<String> partDropdown = new JComboBox<>(new String[]{"Table Top Color", "Legs Co"});
partDropdown.setPreferredSize(new Dimension( width: 150, height: 30)); // Adjust dropdown men

```

```
// Create a dropdown menu for selecting colors for each part
JComboBox<String> partDropdown = new JComboBox<String>[]{"Table Top Color", "Legs Color", "Shading", "Table Style", "Save Page", "Settings Page", "Logout Page", "Order Page", "Preview Panel", "Privacy Policy Page", "Register Page", "Shading 1", "Shading 2", "Table Style 1", "Table Style 2", "Table Style 3", "Table Style 4", "Terms Conditions Page"};
partDropdown.setPreferredSize(new Dimension(150, 30)); // Adjust dropdown size
partDropdown.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JComboBox<String> comboBox = (JComboBox<String>) e.getSource();
        int selectedIndex = comboBox.getSelectedIndex();
        Color selectedColor = JColorChooser.showDialog(null, null, "Choose Color");
        if (selectedColor != null) {
            partColors[selectedIndex] = selectedColor;
            panel.repaint(); // Repaint the GLJPanel to reflect color changes
        }
    }
});
bottomPanel.add(partDropdown, BorderLayout.CENTER);

JSlider scaleSlider = new JSlider(JSlider.HORIZONTAL, 0, 200, 100);
scaleSlider.setMajorTickSpacing(10);
scaleSlider.setPaintTicks(true);
scaleSlider.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        JSlider source = (JSlider) e.getSource();
        if (!source.getValueIsAdjusting()) {
            int value = source.getValue();
            scaleFactor = value / 100;
        }
    }
});
JPanel adjustSizePanel = new JPanel();
JLabel adjustSizeLabel = new JLabel("Adjust size:");
adjustSizePanel.add(adjustSizeLabel);
adjustSizePanel.add(scaleSlider);
bottomPanel.add(adjustSizePanel, BorderLayout.EAST);

// Add "Back" button
 JButton backButton = new JButton("Back");
backButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Navigate to ChairStyleTiles
        JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(TableStyle1.this);
        frame.getContentPane().removeAll();
        frame.getContentPane().add(new ChairStyleTiles());
        frame.revalidate();
    }
});
```

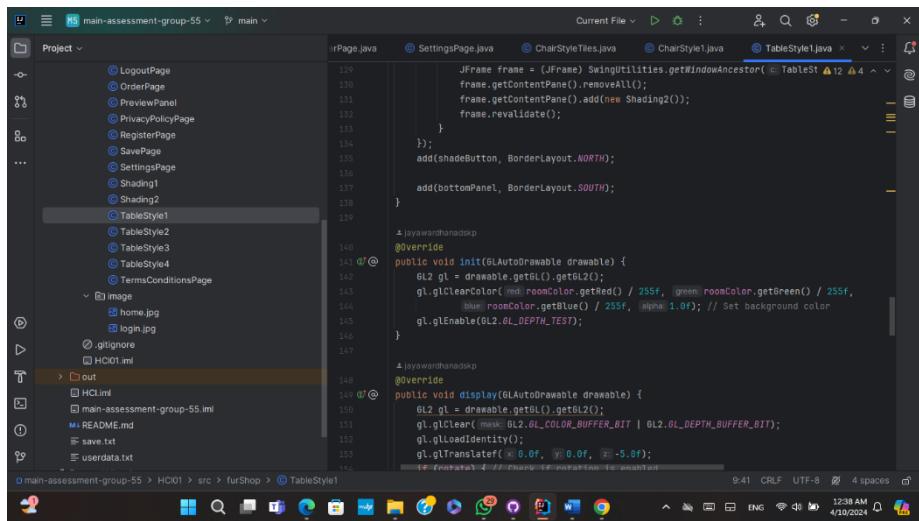
```
public void stateChanged(ChangeEvent e) {
    JSlider source = (JSlider) e.getSource();
    if (!source.getValueIsAdjusting()) {
        int value = source.getValue();
        scaleFactor = value / 100;
    }
};

// Add "Back" button
 JButton backButton = new JButton("Back");
backButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Navigate to ChairStyleTiles
        JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(TableStyle1.this);
        frame.getContentPane().removeAll();
        frame.getContentPane().add(new ChairStyleTiles());
        frame.revalidate();
    }
});
```

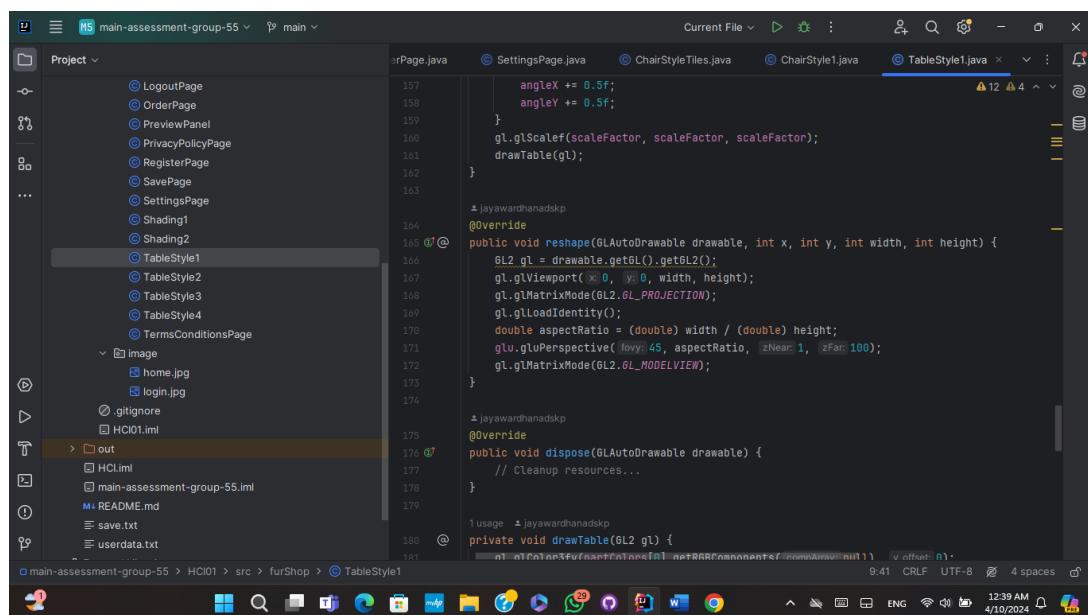
```
bottomPanel.add(backButton, BorderLayout.NORTH);

// Add "Next" button
 JButton nextButton = new JButton("Next");
nextButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Navigate to Save page
        JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(TableStyle1.this);
        frame.getContentPane().removeAll();
        frame.getContentPane().add(new SavePage());
        frame.revalidate();
    }
});
bottomPanel.add(nextButton, BorderLayout.SOUTH);

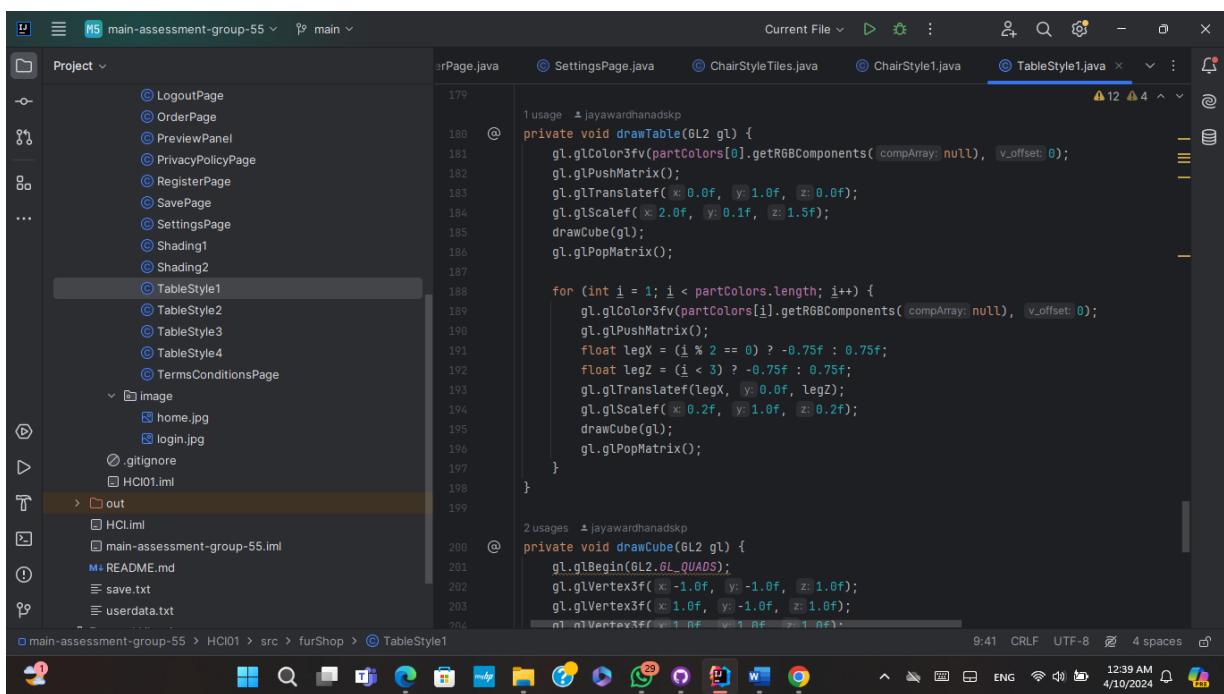
// Add "Shade" button
 JButton shadeButton = new JButton("Shade");
shadeButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Navigate to Shading
        JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(TableStyle1.this);
        frame.getContentPane().removeAll();
        frame.getContentPane().add(new Shading());
        frame.revalidate();
    }
});
```



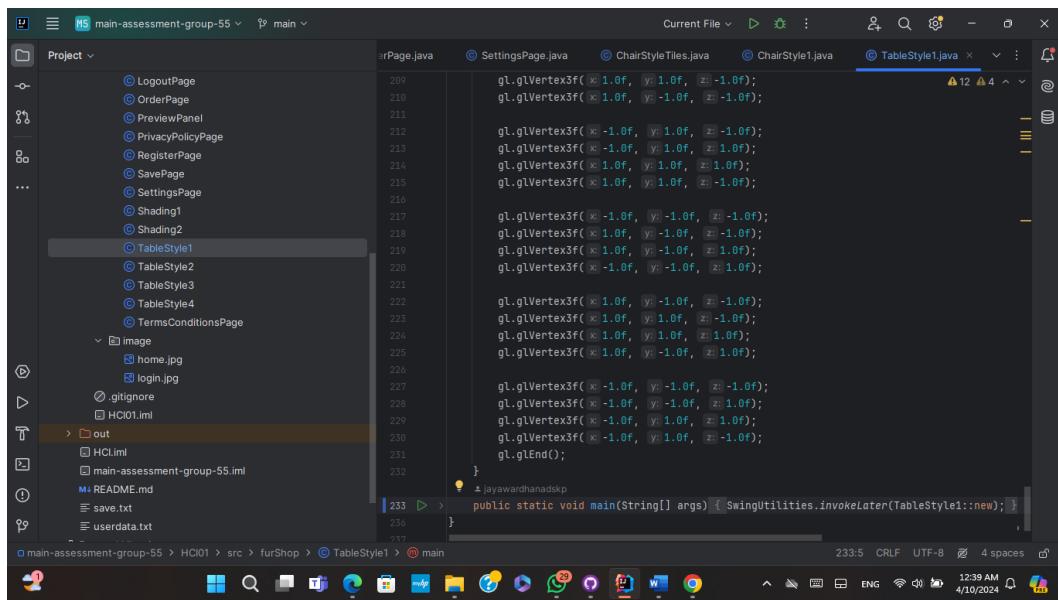
```
129     JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
130     frame.getContentPane().removeAll();
131     frame.getContentPane().add(new Shading2());
132     frame.revalidate();
133   }
134 }
135 add(shadeButton, BorderLayout.NORTH);
136
137 add(bottomPanel, BorderLayout.SOUTH);
138
139
140 /**
141  * @param drawable
142  */
143 public void init(GLAutoDrawable drawable) {
144     GL2 gl = drawable.getGL().getGL2();
145     gl.gClearColor(red: roomColor.getRed() / 255f, green: roomColor.getGreen() / 255f,
146                   blue: roomColor.getBlue() / 255f, alpha: 1.0f); // Set background color
147     gl.gLEnable(GL2.GL_DEPTH_TEST);
148
149 /**
150  * @param drawable
151  */
152 public void display(GLAutoDrawable drawable) {
153     GL2 gl = drawable.getGL().getGL2();
154     gl.gClear(mask: GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);
155     gl.gLoadIdentity();
156     gl.gTranslatef(x: 0.0f, y: 0.0f, z: -5.0f);
157
158     angleX += 0.5f;
159     angleY += 0.5f;
160
161     gl.gScalef(scaleFactor, scaleFactor, scaleFactor);
162     drawTable(gl);
163
164 /**
165  * @param drawable
166  * @param x
167  * @param y
168  * @param width
169  * @param height
170  */
171 public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
172     GL2 gl = drawable.getGL().getGL2();
173     gl.gViewport(x: 0, y: 0, width, height);
174     gl.gMatrixMode(GL2.GL_PROJECTION);
175     gl.gLoadIdentity();
176     double aspectRatio = (double) width / (double) height;
177     glu.gluPerspective(fov: 45, aspectRatio, zNear: 1, zFar: 100);
178     gl.gMatrixMode(GL2.GL_MODELVIEW);
179
180 /**
181  * @param drawable
182  */
183 public void dispose(GLAutoDrawable drawable) {
184     // Cleanup resources...
185 }
```



```
157     angleX += 0.5f;
158     angleY += 0.5f;
159   }
160   gl.gScalef(scaleFactor, scaleFactor, scaleFactor);
161   drawTable(gl);
162
163 /**
164  * @param drawable
165  * @param x
166  * @param y
167  * @param width
168  * @param height
169  */
170 public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
171     GL2 gl = drawable.getGL().getGL2();
172     gl.gViewport(x: 0, y: 0, width, height);
173     gl.gMatrixMode(GL2.GL_PROJECTION);
174     gl.gLoadIdentity();
175     double aspectRatio = (double) width / (double) height;
176     glu.gluPerspective(fov: 45, aspectRatio, zNear: 1, zFar: 100);
177     gl.gMatrixMode(GL2.GL_MODELVIEW);
178
179 /**
180  * @param drawable
181  */
182 public void dispose(GLAutoDrawable drawable) {
183     // Cleanup resources...
184 }
```

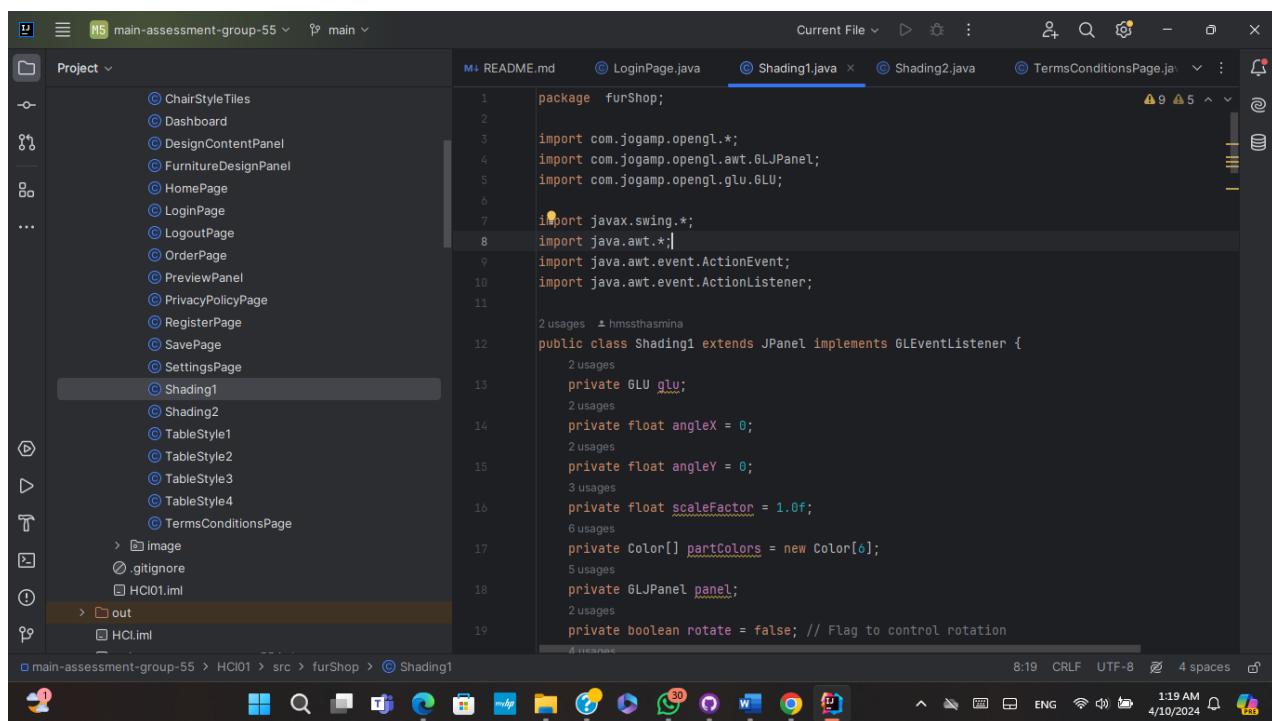


```
179
180 /**
181  * @param drawable
182  */
183 private void drawTable(GL2 gl) {
184     gl.gColor3fv(partColors[0].getRGBComponents(compArray: null), v_offset: 0);
185     gl.gPushMatrix();
186     gl.gTranslate(x: 0.0f, y: 1.0f, z: 0.0f);
187     gl.gScalef(x: 2.0f, y: 0.1f, z: 1.5f);
188     drawCube(gl);
189     gl.gPopMatrix();
190
191     for (int i = 1; i < partColors.length; i++) {
192         gl.gColor3fv(partColors[i].getRGBComponents(compArray: null), v_offset: 0);
193         gl.gPushMatrix();
194         float legX = (i % 2 == 0) ? -0.75f : 0.75f;
195         float legZ = (i < 3) ? -0.75f : 0.75f;
196         gl.gTranslate(x: legX, y: 0.0f, z: legZ);
197         gl.gScalef(x: 0.2f, y: 1.0f, z: 0.2f);
198         drawCube(gl);
199         gl.gPopMatrix();
200
201     }
202
203 /**
204  * @param drawable
205  */
206 private void drawCube(GL2 gl) {
207     gl.gBegin(GL2.GL_QUADS);
208     gl.gVertex3f(x: -1.0f, y: -1.0f, z: 1.0f);
209     gl.gVertex3f(x: 1.0f, y: -1.0f, z: 1.0f);
210     gl.gVertex3f(x: 1.0f, y: 1.0f, z: 1.0f);
211     gl.gVertex3f(x: -1.0f, y: 1.0f, z: 1.0f);
212 }
```

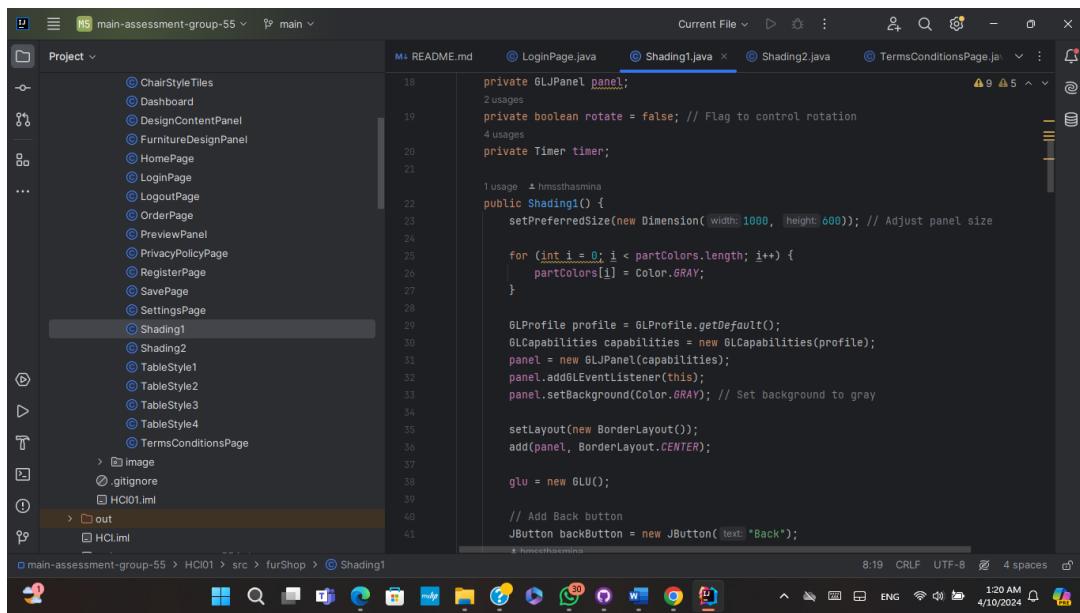


```
Current File ▾ ⌂ main-assessment-group-55 ▾ main ▾ Project ▾ LogoutPage.java SettingsPage.java ChairStyleTiles.java ChairStyle1.java TableStyle1.java
209 gl.glVertex3f( < 1.0f, y: 1.0f, z: -1.0f);
210 gl.glVertex3f( < 1.0f, y: -1.0f, z: -1.0f);
211
212 gl.glVertex3f( < -1.0f, y: 1.0f, z: -1.0f);
213 gl.glVertex3f( < -1.0f, y: -1.0f, z: -1.0f);
214 gl.glVertex3f( < 1.0f, y: 1.0f, z: 1.0f);
215 gl.glVertex3f( < 1.0f, y: -1.0f, z: 1.0f);
216
217 gl.glVertex3f( < -1.0f, y: -1.0f, z: -1.0f);
218 gl.glVertex3f( < 1.0f, y: -1.0f, z: -1.0f);
219 gl.glVertex3f( < 1.0f, y: -1.0f, z: 1.0f);
220 gl.glVertex3f( < -1.0f, y: -1.0f, z: 1.0f);
221
222 gl.glVertex3f( < 1.0f, y: 1.0f, z: -1.0f);
223 gl.glVertex3f( < 1.0f, y: 1.0f, z: 1.0f);
224 gl.glVertex3f( < -1.0f, y: 1.0f, z: 1.0f);
225 gl.glVertex3f( < -1.0f, y: 1.0f, z: -1.0f);
226
227 gl.glVertex3f( < -1.0f, y: -1.0f, z: -1.0f);
228 gl.glVertex3f( < -1.0f, y: -1.0f, z: 1.0f);
229 gl.glVertex3f( < -1.0f, y: 1.0f, z: 1.0f);
230 gl.glVertex3f( < -1.0f, y: 1.0f, z: -1.0f);
231 gl.glEnd();
232
233 public static void main(String[] args) { SwingUtilities.invokeLater(TableStyle1::new); }
234 }
```

➤ Shading1 java class.



```
Current File ▾ ⌂ main-assessment-group-55 ▾ main ▾ Project ▾ README.md LoginPage.java Shading1.java Shading2.java TermsConditionsPage.java
1 package furShop;
2
3 import com.jogamp.opengl.*;
4 import com.jogamp.opengl.awt.GLJPanel;
5 import com.jogamp.opengl.glu.GLU;
6
7 import javax.swing.*;
8 import java.awt.*;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11
12 2 usages ↗ hmssthasmina
13 public class Shading1 extends JPanel implements GLEventListener {
14     2 usages
15     private GLU glu;
16     2 usages
17     private float angleX = 0;
18     2 usages
19     private float angleY = 0;
20     3 usages
21     private float scaleFactor = 1.0f;
22     6 usages
23     private Color[] partColors = new Color[6];
24     5 usages
25     private GLJPanel panel;
26     2 usages
27     private boolean rotate = false; // Flag to control rotation
28
29 }
```



```
private GLJPanel panel;
2 usages
private boolean rotate = false; // Flag to control rotation
4 usages
private Timer timer;

1 usage  ↳ hmsshasmina
public Shading1() {
    setPreferredSize(new Dimension( width: 1000, height: 600)); // Adjust panel size

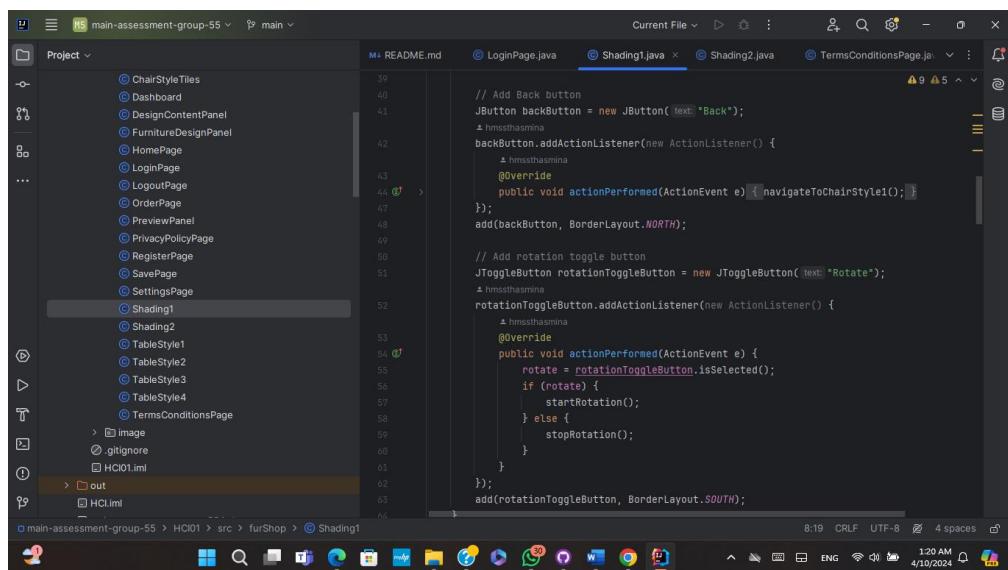
    for (int i = 0; i < partColors.length; i++) {
        partColors[i] = Color.GRAY;
    }

    GLProfile profile = GLProfile.getDefault();
    GLCapabilities capabilities = new GLCapabilities(profile);
    panel = new GLJPanel(capabilities);
    panel.addGLEventListener(this);
    panel.setBackground(Color.GRAY); // Set background to gray

    setLayout(new BorderLayout());
    add(panel, BorderLayout.CENTER);

    glu = new GLU();

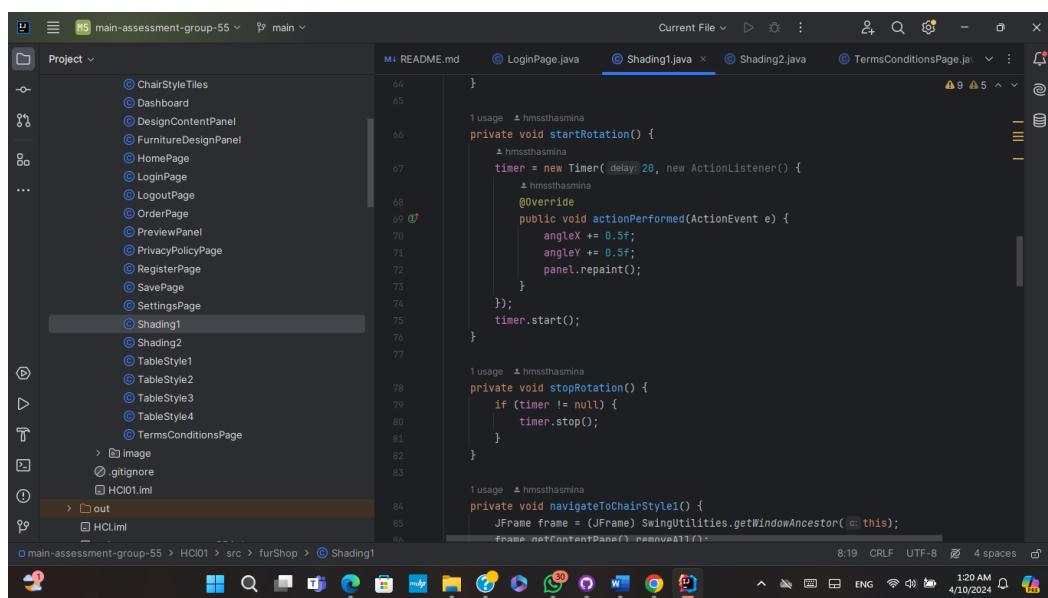
    // Add Back button
    JButton backButton = new JButton( text: "Back");
    
```



```
// Add Back button
JButton backButton = new JButton( text: "Back");
↳ hmsshasmina
backButton.addActionListener(new ActionListener() {
    ↳ hmsshasmina
    @Override
    public void actionPerformed(ActionEvent e) { navigateToChairStyle1(); }
});
add(backButton, BorderLayout.NORTH);

// Add rotation toggle button
JToggleButton rotationToggleButton = new JToggleButton( text: "Rotate");
↳ hmsshasmina
rotationToggleButton.addActionListener(new ActionListener() {
    ↳ hmsshasmina
    @Override
    public void actionPerformed(ActionEvent e) {
        rotate = rotationToggleButton.isSelected();
        if (rotate) {
            startRotation();
        } else {
            stopRotation();
        }
    }
});
add(rotationToggleButton, BorderLayout.SOUTH);

```



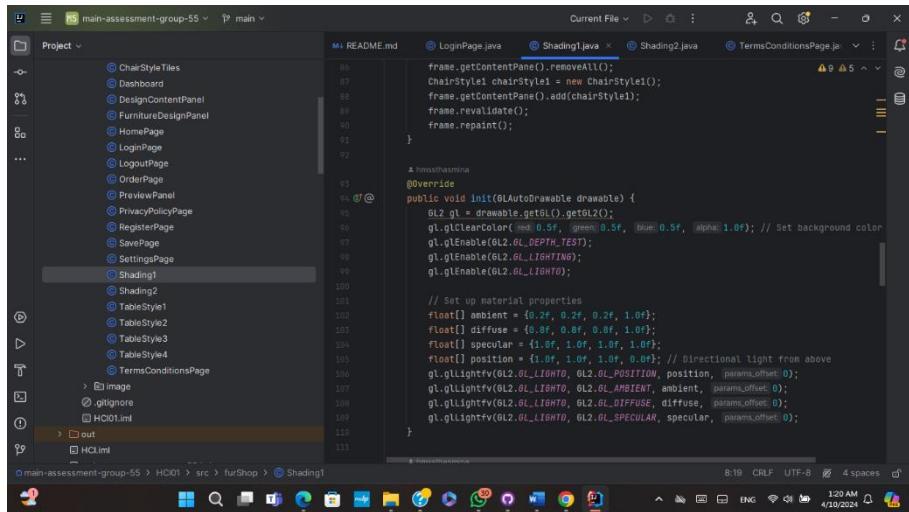
```
}
}

1 usage  ↳ hmsshasmina
private void startRotation() {
    ↳ hmsshasmina
    timer = new Timer( delay: 20, new ActionListener() {
        ↳ hmsshasmina
        @Override
        public void actionPerformed(ActionEvent e) {
            angleX += 0.5f;
            angleY += 0.5f;
            panel.repaint();
        }
    });
    timer.start();
}

1 usage  ↳ hmsshasmina
private void stopRotation() {
    if (timer != null) {
        timer.stop();
    }
}

1 usage  ↳ hmsshasmina
private void navigateToChairStyle1() {
    JFrame frame = (JFrame) SwingUtilities.getWindowAncestor( c: this );
    frame.getContentPane().removeAll();
}

```



```
frame.getContentPane().removeAll();
ChairStyle chairStyle1 = new ChairStyle1();
frame.getContentPane().add(chairStyle1);
frame.revalidate();
frame.repaint();

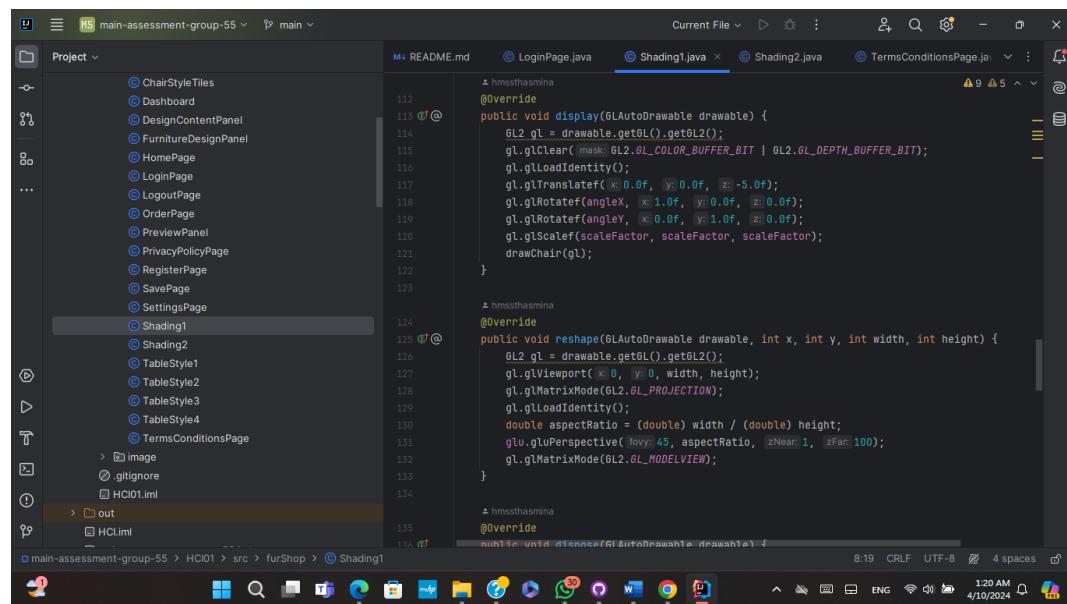
@Override
public void init(GLAutoDrawable drawable) {
    GL2 gl = drawable.getGL();
    gl.glClearColor(red: 0.5f, green: 0.5f, blue: 0.5f, alpha: 1.0f); // Set background color
    gl.gLEnable(GL2.GL_DEPTH_TEST);
    gl.gLEnable(GL2.GL_LIGHTING);
    gl.gLEnable(GL2.GL_TEXTURE);

    // Set up material properties
    float[] ambient = {0.2f, 0.2f, 0.2f, 1.0f};
    float[] diffuse = {0.9f, 0.9f, 0.9f, 1.0f};
    float[] specular = {1.0f, 1.0f, 1.0f, 1.0f};
    float[] position = {1.0f, 1.0f, 1.0f, 0.0f}; // Directional light from above
    gl.gLightfv(GL2.GL_LIGHT0, GL2.GL_POSITION, position, params_offset: 0);
    gl.gLightfv(GL2.GL_LIGHT0, GL2.GL_AMBIENT, ambient, params_offset: 0);
    gl.gLightfv(GL2.GL_LIGHT0, GL2.GL_DIFFUSE, diffuse, params_offset: 0);
    gl.gLightfv(GL2.GL_LIGHT0, GL2.GL_SPECULAR, specular, params_offset: 0);
}

// hmsshasmina
@Override
public void display(GLAutoDrawable drawable) {
    GL2 gl = drawable.getGL();
    gl.gLClear(mask: GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);
    gl.gLLoadIdentity();
    gl.gLTranslatef(x: 0.0f, y: 0.0f, z: -5.0f);
    gl.gLRotatef(angleX, x: 1.0f, y: 0.0f, z: 0.0f);
    gl.gLRotatef(angleY, x: 0.0f, y: 1.0f, z: 0.0f);
    gl.gLScalef(scaleFactor, scaleFactor, scaleFactor);
    drawChair(gl);
}

// hmsshasmina
@Override
public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
    GL2 gl = drawable.getGL();
    gl.glViewport(x: 0, y: 0, width, height);
    gl.gLMatrixMode(GL2.GL_PROJECTION);
    gl.gLLoadIdentity();
    double aspectRatio = (double) width / (double) height;
    glu.gluPerspective(fov: 45, aspectRatio, zNear: 1, zFar: 100);
    gl.gLMatrixMode(GL2.GL_MODELVIEW);
}

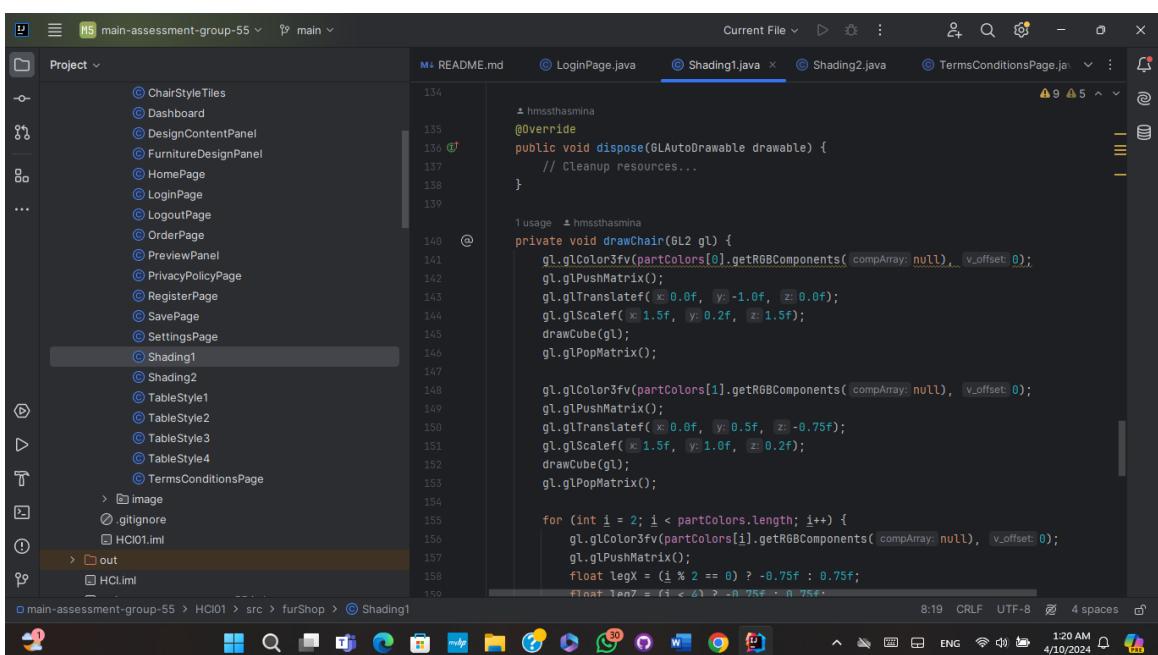
// hmsshasmina
@Override
public void dispose(GLAutoDrawable drawable) {
    // Cleanup resources...
}
```



```
112
113 @Override
114 public void display(GLAutoDrawable drawable) {
115     GL2 gl = drawable.getGL();
116     gl.gLClear(mask: GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);
117     gl.gLLoadIdentity();
118     gl.gLTranslatef(x: 0.0f, y: 0.0f, z: -5.0f);
119     gl.gLRotatef(angleX, x: 1.0f, y: 0.0f, z: 0.0f);
120     gl.gLRotatef(angleY, x: 0.0f, y: 1.0f, z: 0.0f);
121     gl.gLScalef(scaleFactor, scaleFactor, scaleFactor);
122     drawChair(gl);
123 }

124
125 @Override
126 public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
127     GL2 gl = drawable.getGL();
128     gl.glViewport(x: 0, y: 0, width, height);
129     gl.gLMatrixMode(GL2.GL_PROJECTION);
130     gl.gLLoadIdentity();
131     double aspectRatio = (double) width / (double) height;
132     glu.gluPerspective(fov: 45, aspectRatio, zNear: 1, zFar: 100);
133     gl.gLMatrixMode(GL2.GL_MODELVIEW);
134 }

135
136 @Override
137 public void dispose(GLAutoDrawable drawable) {
138     // Cleanup resources...
139 }
```

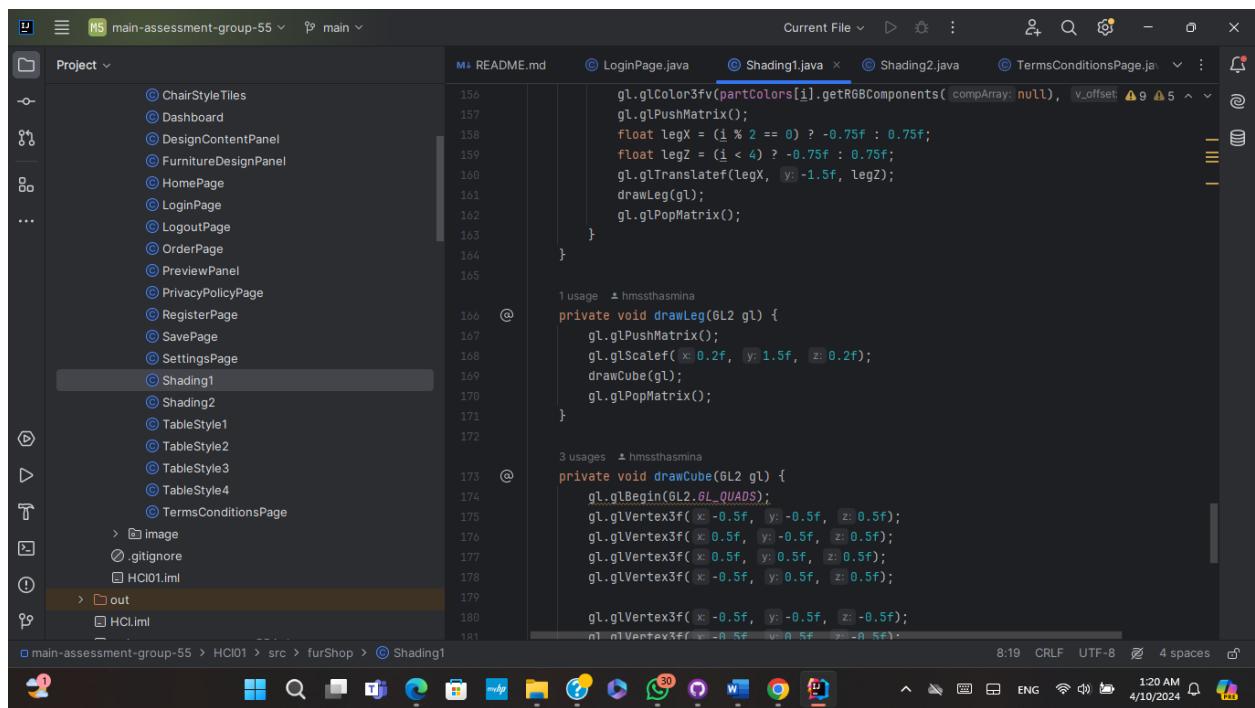


```
134
135
136 @Override
137 public void display(GLAutoDrawable drawable) {
138     // Cleanup resources...
139 }

140
141 private void drawChair(GL2 gl) {
142     gl.glColor3fv(partColors[0].getRGBComponents(compArray: null), v_offset: 0);
143     gl.gLPushMatrix();
144     gl.gLTranslatef(x: 0.0f, y: -1.0f, z: 0.0f);
145     gl.gLScalef(x: 1.5f, y: 0.2f, z: 1.5f);
146     drawCube(gl);
147     gl.gLPopMatrix();

148     gl.glColor3fv(partColors[1].getRGBComponents(compArray: null), v_offset: 0);
149     gl.gLPushMatrix();
150     gl.gLTranslatef(x: 0.0f, y: 0.5f, z: -0.75f);
151     gl.gLScalef(x: 1.5f, y: 1.0f, z: 0.2f);
152     drawCube(gl);
153     gl.gLPopMatrix();

154     for (int i = 2; i < partColors.length; i++) {
155         gl.glColor3fv(partColors[i].getRGBComponents(compArray: null), v_offset: 0);
156         gl.gLPushMatrix();
157         float legX = (i % 2 == 0) ? -0.75f : 0.75f;
158         float legZ = (i < 4) ? -0.75f : 0.75f;
```



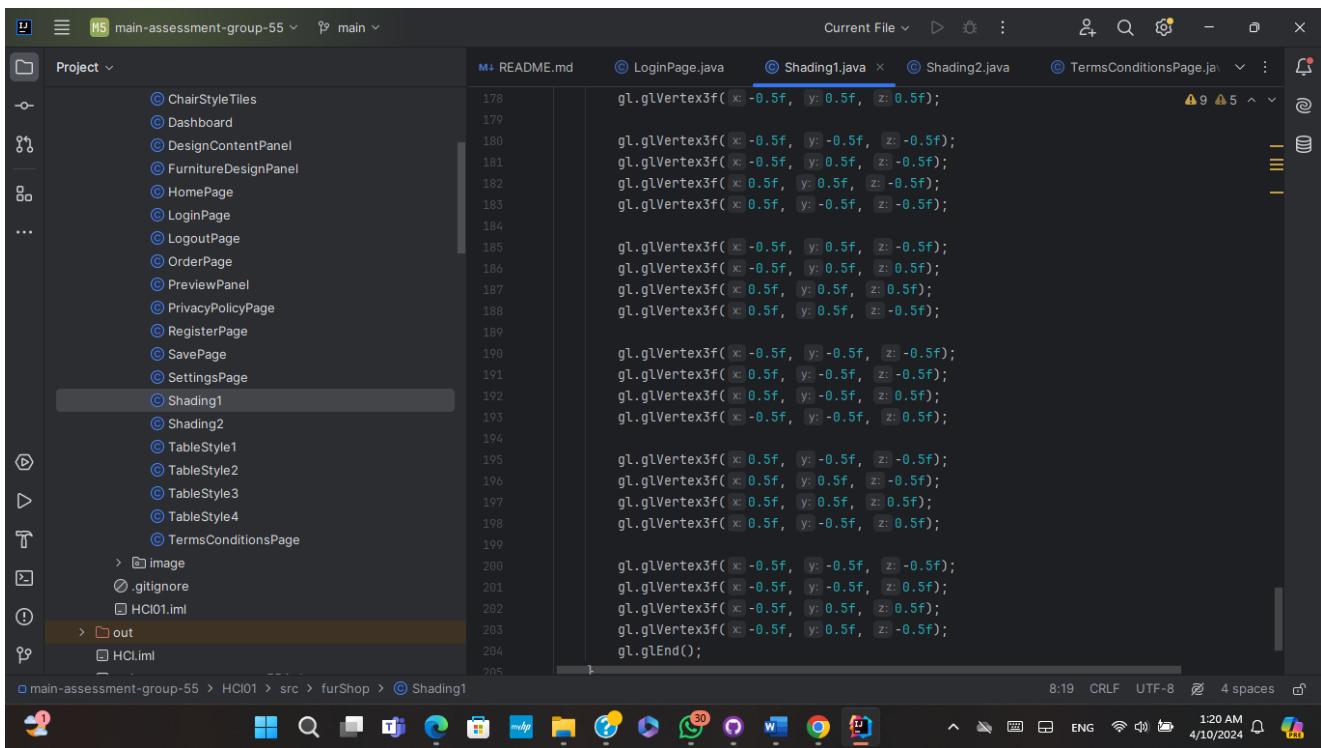
```
gl.glColor3fv(partColors[i].getRGBComponents(compArray: null), v_offset: 9 5 ^ v);
gl.gPushMatrix();
float legX = (i % 2 == 0) ? -0.75f : 0.75f;
float legZ = (i < 4) ? -0.75f : 0.75f;
gl.gLTranslatef(legX, y: -1.5f, legZ);
drawLeg(gl);
gl.gPopMatrix();

}

1 usage ± hmssthasmina
private void drawLeg(GL2 gl) {
    gl.gPushMatrix();
    gl.gScalef(x: 0.2f, y: 1.5f, z: 0.2f);
    drawCube(gl);
    gl.gPopMatrix();
}

3 usages ± hmssthasmina
private void drawCube(GL2 gl) {
    gl.gBegin(GL_QUADS);
    gl.gVertex3f(x: -0.5f, y: -0.5f, z: 0.5f);
    gl.gVertex3f(x: 0.5f, y: -0.5f, z: 0.5f);
    gl.gVertex3f(x: 0.5f, y: 0.5f, z: 0.5f);
    gl.gVertex3f(x: -0.5f, y: 0.5f, z: 0.5f);

    gl.gVertex3f(x: -0.5f, y: -0.5f, z: -0.5f);
    gl.gVertex3f(x: 0.5f, y: -0.5f, z: -0.5f);
    gl.gVertex3f(x: 0.5f, y: 0.5f, z: -0.5f);
    gl.gVertex3f(x: -0.5f, y: 0.5f, z: -0.5f);
}
```



```
gl.gVertex3f(x: -0.5f, y: 0.5f, z: 0.5f);
gl.gVertex3f(x: -0.5f, y: -0.5f, z: 0.5f);
gl.gVertex3f(x: 0.5f, y: 0.5f, z: 0.5f);
gl.gVertex3f(x: 0.5f, y: -0.5f, z: 0.5f);

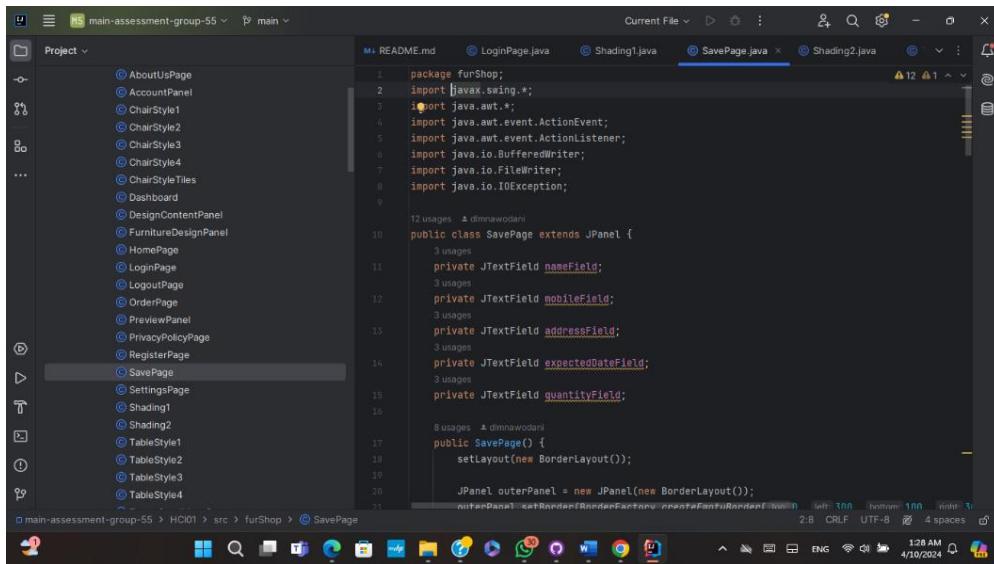
gl.gVertex3f(x: -0.5f, y: 0.5f, z: -0.5f);
gl.gVertex3f(x: 0.5f, y: 0.5f, z: -0.5f);
gl.gVertex3f(x: 0.5f, y: -0.5f, z: -0.5f);
gl.gVertex3f(x: -0.5f, y: -0.5f, z: -0.5f);

gl.gVertex3f(x: 0.5f, y: 0.5f, z: 0.5f);
gl.gVertex3f(x: 0.5f, y: -0.5f, z: 0.5f);
gl.gVertex3f(x: -0.5f, y: 0.5f, z: -0.5f);
gl.gVertex3f(x: -0.5f, y: -0.5f, z: -0.5f);

gl.gVertex3f(x: 0.5f, y: 0.5f, z: -0.5f);
gl.gVertex3f(x: 0.5f, y: -0.5f, z: -0.5f);
gl.gVertex3f(x: -0.5f, y: 0.5f, z: 0.5f);
gl.gVertex3f(x: -0.5f, y: -0.5f, z: 0.5f);

gl.gEnd();
```

➤ Save page java class.



```
package furShop;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
```

```
public class SavePage extends JPanel {
    private JTextField nameField;
    private JTextField mobileField;
    private JTextField addressField;
    private JTextField expectedDateField;
    private JTextField quantityField;
```

```
public SavePage() {
    setLayout(new BorderLayout());
    JPanel outerPanel = new JPanel(new BorderLayout());
    outerPanel.setBorder(BorderFactory.createEmptyBorder(10, 100, 100, 10));
    outerPanel.setLayout(new GridLayout(1, 1));
    outerPanel.add(this);
    this.setLayout(new BorderLayout());
    JPanel formPanel = new JPanel(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridx = 0;
    gbc.gridy = 0;
    gbc.anchor = GridBagConstraints.WEST;
    gbc.insets = new Insets(5, 10, 5, 10);

    JLabel nameLabel = new JLabel("Name:");
    formPanel.add(nameLabel, gbc);

    gbc.gridx++;
    JLabel mobileLabel = new JLabel("Mobile:");
    formPanel.add(mobileLabel, gbc);

    gbc.gridx++;
    JLabel addressLabel = new JLabel("Address:");
    formPanel.add(addressLabel, gbc);

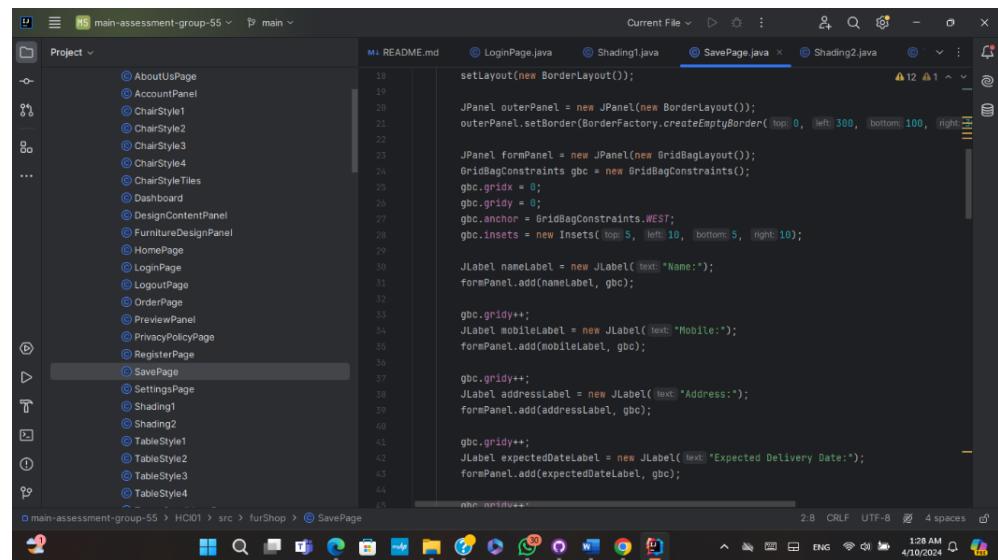
    gbc.gridx++;
    JLabel expectedDateLabel = new JLabel("Expected Delivery Date:");
    formPanel.add(expectedDateLabel, gbc);

    gbc.gridx++;
    JTextField nameField = new JTextField(20);
    formPanel.add(nameField, gbc);

    gbc.gridx++;
    JTextField mobileField = new JTextField(20);
    formPanel.add(mobileField, gbc);

    gbc.gridx++;
    JTextField addressField = new JTextField(20);
    formPanel.add(addressField, gbc);

    gbc.gridx++;
    JTextField expectedDateField = new JTextField(20);
    formPanel.add(expectedDateField, gbc);
```



```
setLayout(new BorderLayout());
JPanel outerPanel = new JPanel(new BorderLayout());
outerPanel.setBorder(BorderFactory.createEmptyBorder(10, 100, 100, 10));
outerPanel.setLayout(new GridLayout(1, 1));
outerPanel.add(this);
this.setLayout(new BorderLayout());
JPanel formPanel = new JPanel(new GridBagLayout());
GridBagConstraints gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.WEST;
gbc.insets = new Insets(5, 10, 5, 10);

JLabel nameLabel = new JLabel("Name:");
formPanel.add(nameLabel, gbc);

gbc.gridx++;
JLabel mobileLabel = new JLabel("Mobile:");
formPanel.add(mobileLabel, gbc);

gbc.gridx++;
JLabel addressLabel = new JLabel("Address:");
formPanel.add(addressLabel, gbc);

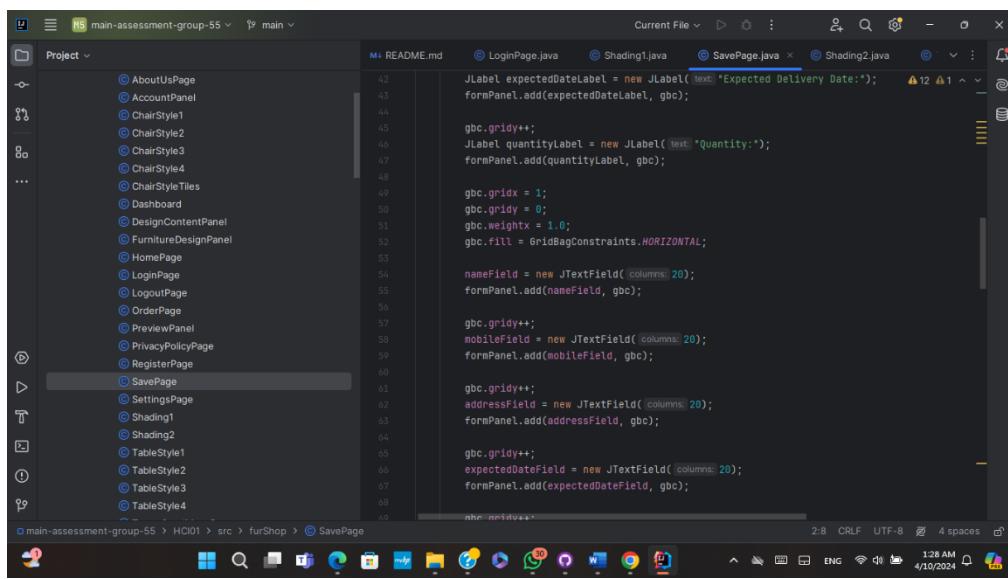
gbc.gridx++;
JLabel expectedDateLabel = new JLabel("Expected Delivery Date:");
formPanel.add(expectedDateLabel, gbc);

gbc.gridx++;
JTextField nameField = new JTextField(20);
formPanel.add(nameField, gbc);

gbc.gridx++;
JTextField mobileField = new JTextField(20);
formPanel.add(mobileField, gbc);

gbc.gridx++;
JTextField addressField = new JTextField(20);
formPanel.add(addressField, gbc);

gbc.gridx++;
JTextField expectedDateField = new JTextField(20);
formPanel.add(expectedDateField, gbc);
```



```
41 | Page
```

```
package furShop;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
```

```
public class SavePage extends JPanel {
    private JTextField nameField;
    private JTextField mobileField;
    private JTextField addressField;
    private JTextField expectedDateField;
    private JTextField quantityField;
```

```
public SavePage() {
    setLayout(new BorderLayout());
    JPanel outerPanel = new JPanel(new BorderLayout());
    outerPanel.setBorder(BorderFactory.createEmptyBorder(10, 100, 100, 10));
    outerPanel.setLayout(new GridLayout(1, 1));
    outerPanel.add(this);
    this.setLayout(new BorderLayout());
    JPanel formPanel = new JPanel(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridx = 0;
    gbc.gridy = 0;
    gbc.anchor = GridBagConstraints.WEST;
    gbc.insets = new Insets(5, 10, 5, 10);

    JLabel nameLabel = new JLabel("Name:");
    formPanel.add(nameLabel, gbc);

    gbc.gridx++;
    JLabel mobileLabel = new JLabel("Mobile:");
    formPanel.add(mobileLabel, gbc);

    gbc.gridx++;
    JLabel addressLabel = new JLabel("Address:");
    formPanel.add(addressLabel, gbc);

    gbc.gridx++;
    JLabel expectedDateLabel = new JLabel("Expected Delivery Date:");
    formPanel.add(expectedDateLabel, gbc);

    gbc.gridx++;
    JTextField nameField = new JTextField(20);
    formPanel.add(nameField, gbc);

    gbc.gridx++;
    JTextField mobileField = new JTextField(20);
    formPanel.add(mobileField, gbc);

    gbc.gridx++;
    JTextField addressField = new JTextField(20);
    formPanel.add(addressField, gbc);

    gbc.gridx++;
    JTextField expectedDateField = new JTextField(20);
    formPanel.add(expectedDateField, gbc);
```

Project: main-assessment-group-55

File: SavePage.java

```
gbc.gridx++;  
quantityField = new JTextField( columns: 20);  
formPanel.add(quantityField, gbc);  
  
gbc.gridx = 0;  
gbc.gridy++;  
gbc.gridwidth = 2;  
gbc.fill = GridBagConstraints.NONE;  
gbc.anchor = GridBagConstraints.CENTER;  
  
JButton saveButton = new JButton( text: "Save");  
saveButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        String name = nameField.getText();  
        String mobile = mobileField.getText();  
        String address = addressField.getText();  
        String expectedDate = expectedDateField.getText();  
        String quantityText = quantityField.getText();  
  
        if (name.isEmpty() || mobile.isEmpty() || address.isEmpty() || expectedDate.isEmpty())  
            JOptionPane.showMessageDialog( parentComponent: null, message: "Please fill in all fields");  
        return;  
    }  
});  
  
try {  
    int quantity = Integer.parseInt(quantityText);  
  
    String orderId = saveDesign(name, mobile, address, expectedDate, quantity);  
    String details = orderId + ", " + name + ", " + mobile + ", " + address + ", " + quantity;  
  
    saveDetailsToFile(details);  
  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Design saved successfully");  
} catch (NumberFormatException ex) {  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Invalid quantity");  
} catch (IOException ex) {  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Failed to save design");  
}  
  
JButton viewOrderButton = new JButton( text: "View Order");  
viewOrderButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Show order page  
        JFrame frame = new JFrame( title: "Order Details");  
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
        frame.getContentPane().add(new OrderPage());  
        frame.setExtendedState(JFrame.MAXIMIZED_BOTH); // Maximize the frame  
        frame.setVisible(true);  
    }  
});  
outerPanel.add(formPanel, BorderLayout.CENTER);  
outerPanel.add(viewOrderButton, BorderLayout.SOUTH);  
add(outerPanel, BorderLayout.CENTER);  
  
1 usage  
private String saveDesign(String name, String mobile, String address, String expectedDate, int quantity)  
    // Placeholder for saving logic  
    // Generate a unique order ID, for example, using a combination of current time and a random number  
    return "ORD" + System.currentTimeMillis();  
}
```

Project: main-assessment-group-55

File: SavePage.java

```
if (name.isEmpty() || mobile.isEmpty() || address.isEmpty() || expectedDate.isEmpty())  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Please fill in all fields");  
return;  
  
try {  
    int quantity = Integer.parseInt(quantityText);  
  
    String orderId = saveDesign(name, mobile, address, expectedDate, quantity);  
    String details = orderId + ", " + name + ", " + mobile + ", " + address + ", " + quantity;  
  
    saveDetailsToFile(details);  
  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Design saved successfully");  
} catch (NumberFormatException ex) {  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Invalid quantity");  
} catch (IOException ex) {  
    JOptionPane.showMessageDialog( parentComponent: null, message: "Failed to save design");  
}  
  
JButton viewOrderButton = new JButton( text: "View Order");  
viewOrderButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Show order page  
        JFrame frame = new JFrame( title: "Order Details");  
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
        frame.getContentPane().add(new OrderPage());  
        frame.setExtendedState(JFrame.MAXIMIZED_BOTH); // Maximize the frame  
        frame.setVisible(true);  
    }  
});  
outerPanel.add(formPanel, BorderLayout.CENTER);  
outerPanel.add(viewOrderButton, BorderLayout.SOUTH);  
add(outerPanel, BorderLayout.CENTER);  
  
1 usage  
private String saveDesign(String name, String mobile, String address, String expectedDate, int quantity)  
    // Placeholder for saving logic  
    // Generate a unique order ID, for example, using a combination of current time and a random number  
    return "ORD" + System.currentTimeMillis();  
}
```

Project: main-assessment-group-55

File: SavePage.java

```
JButton viewOrderButton = new JButton( text: "View Order");  
viewOrderButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Show order page  
        JFrame frame = new JFrame( title: "Order Details");  
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
        frame.getContentPane().add(new OrderPage());  
        frame.setExtendedState(JFrame.MAXIMIZED_BOTH); // Maximize the frame  
        frame.setVisible(true);  
    }  
});  
outerPanel.add(formPanel, BorderLayout.CENTER);  
outerPanel.add(viewOrderButton, BorderLayout.SOUTH);  
add(outerPanel, BorderLayout.CENTER);  
  
1 usage  
private String saveDesign(String name, String mobile, String address, String expectedDate, int quantity)  
    // Placeholder for saving logic  
    // Generate a unique order ID, for example, using a combination of current time and a random number  
    return "ORD" + System.currentTimeMillis();  
}
```

The screenshot shows the IntelliJ IDEA interface with the project navigation bar on the left and the code editor on the right. The code editor displays Java code for a `SavePage` class. The code includes methods for saving design details to a file. The status bar at the bottom shows the file path as `main-assessment-group-55 > HCI01 > src > furShop > SavePage > anonymous ActionListener > actionPerformed`, and the current time as 12:00 AM on 4/10/2024.

```
122     frame.setVisible(true);
123 }
124 );
125 outerPanel.add(formPanel, BorderLayout.CENTER);
126 outerPanel.add(viewOrderButton, BorderLayout.SOUTH);
127 add(outerPanel, BorderLayout.CENTER);
128 }

130 @
131 // Placeholder for saving logic
132 // Generate a unique order ID, for example, using a combination of current time and a random number
133 return "ORD" + System.currentTimeMillis();
134 }

136 private void saveDetailsToFile(String details) throws IOException {
137     FileWriter fw = new FileWriter(fileName: "save.txt", append: true);
138     BufferedWriter bw = new BufferedWriter(fw);
139     bw.write(details);
140     bw.newLine();
141     bw.close();
142 }
```

➤ Account panel java class.

The screenshot shows the IntelliJ IDEA interface with the project navigation bar on the left and the code editor on the right. The code editor displays Java code for an `AccountPanel` class. The code includes imports for various Java Swing and IO classes. The status bar at the bottom shows the file path as `main-assessment-group-55 > HCI01 > src > furShop > AccountPanel`, and the current time as 1:39 AM on 4/10/2024.

```
1 package furShop;
2
3 import javax.imageio.ImageIO;
4 import javax.swing.*;
5 import javax.swing.border.EmptyBorder;
6 import java.awt.*;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import java.awt.image.BufferedImage;
10 import java.io.File;
11 import java.io.FileWriter;
12 import java.io.IOException;
13
14 3 usages  dimnawodani +
15 public class AccountPanel extends JPanel {
16     6 usages
17     private JTextField nameField;
18     6 usages
19     private JTextField emailField;
20     6 usages
21     private JTextField mobileField;
22     6 usages
23     private JTextField faxField;
24     10 usages
25     private JButton saveButton;
26     8 usages
27     private JButton backButton;
```

➤ Privacy policy java class.

The screenshot shows a Java IDE interface with the following details:

- Project Explorer:** Shows files like ChairStyleTiles, Dashboard, DesignContentPanel, FurnitureDesignPanel, HomePage, LoginPage, LogoutPage, OrderPage, PreviewPanel, PrivacyPolicyPage (selected), RegisterPage, SavePage, SettingsPage, Shading1, Shading2, TableStyle1, TableStyle2, TableStyle3, TableStyle4, TermsConditionsPage, and HCI01.iml.
- Current File:** PrivacyPolicyPage.java
- Code Content:**

```
1 package furShop;
2
3 import javax.swing.*;
4 import java.awt.*;
5
6 1 usage ▲ Pasindu Jayawardena
7 public class PrivacyPolicyPage extends JPanel {
8     public PrivacyPolicyPage() {
9         setLayout(new BorderLayout());
10        setBackground(Color.WHITE); // Set background color to white
11
12        JTextPane policyTextPane = new JTextPane();
13        policyTextPane.setContentType("text/html"); // Set content type to HTML
14        policyTextPane.setEditable(false); // Make text area read-only
15
16        // Privacy Policy text with HTML formatting
17        String policyText = "<html>" +
18            "<p><b>This Privacy Policy</b> explains how FurShop (\"we\", \"us\", \"our\") col" +
19            "<p><b>1. Information Collection and Use</b><br>" +
20            "We may collect and use personal information to provide and improve our Service." +
21            "<p><b>2. Log Data</b><br>" +
22            "We may also collect information that your browser sends whenever you visit our S" +
23            "<p><b>3. Cookies</b><br>" +
24            "Cookies are files with a small amount of data, which may include an anonymous un" +
25            "<p><b>4. Security</b><br>" +
26            "The security of your personal information is important to us, but remember that" +
27            "<h2>Changes to This Privacy Policy</h2><hr>" +
28
29    1 usage ▲ lakshika
30 }
```
- Bottom Status Bar:** Shows file paths (main-assessment-group-55 > HCI01 > src > furShop), file types (PrivacyPolicyPage.java), encoding (UTF-8), and system status (3:8 CRLF, ENG, 4/10/2024, 1:39 AM).

➤ Term & conditions java class.

The screenshot shows a Java IDE interface with the following details:

- Project Explorer:** Shows files like AccountPanel.java, PrivacyPolicyPage.java, Shading2.java, and TermsConditionsPage.java (selected).
- Current File:** TermsConditionsPage.java
- Code Content:**

```
1 package furShop;
2
3 import javax.swing.*;
4 import java.awt.*;
5
6 1 usage ▲ lakshika
7 public class TermsConditionsPage extends JPanel {
8     public TermsConditionsPage() {
9         setLayout(new BorderLayout());
10        setBackground(Color.WHITE); // Set background color to white
11
12        JTextPane termsTextPane = new JTextPane();
13        termsTextPane.setContentType("text/html"); // Set content type to HTML
14        termsTextPane.setEditable(false); // Make text area read-only
15
16        // Terms and Conditions text with HTML formatting
17        String termsText = "<html>" +
18            "<h1 style='text-align: center; font-size: 24px; font-weight: bold;'>Terms and Co" +
19            "<div style='overflow-y: scroll; max-height: 400px; padding: 10px;'>" +
20            "<ol>" +
21            "<li><strong>Introduction</strong><br>" +
22            "These Terms and Conditions govern your use of [Your Company Name]'s desktop appl" +
23            "<li><strong>Acceptance of Terms</strong><br>" +
24            "By accessing or using the application, you agree to be bound by these Terms and" +
25            "<li><strong>Use of the Application</strong><br>" +
26            "The application is provided on an \"as is\" and \"as available\" basis. [Your Co" +
27            "<hr><hr>Intellectual Property Rights<hr><hr>" +
28
29    1 usage ▲ lakshika
30 }
```
- Bottom Status Bar:** Shows file paths (main-assessment-group-55 > HCI01 > src > furShop), file types (TermsConditionsPage.java), encoding (UTF-8), and system status (3:8 CRLF, ENG, 4/10/2024, 1:39 AM).

➤ About Us java class.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "furShop". The file "AboutUsPage.java" is selected.
- Code Editor:** Displays the Java code for "AboutUsPage.java".
- Toolbars and Status Bar:** Includes standard IntelliJ icons and status information like "3:8 CRLF UTF-8", "4 spaces", and the current time "1:43 AM 4/10/2024".

```
package furShop;
import javax.swing.*;
import java.awt.*;

1 usage  ± NAntonette +1
public class AboutUsPage extends JPanel {
    1 usage  ± NAntonette
    public AboutUsPage() {
        setLayout(new BorderLayout());
        setBackground(new Color( 245,  245,  245));

        // Title Panel
        JPanel titlePanel = new JPanel(new BorderLayout());
        titlePanel.setBackground(new Color( 51,  50,  49));
        JLabel titleLabel = new JLabel( text: "About Us", SwingConstants.CENTER);
        titleLabel.setFont(new Font( name: "Arial", Font.BOLD,  size: 34));
        titleLabel.setForeground(Color.WHITE);
        titlePanel.add(titleLabel, BorderLayout.CENTER);
        add(titlePanel, BorderLayout.NORTH);

        JPanel contentPanel = new JPanel();
        contentPanel.setLayout(new BoxLayout(contentPanel, BoxLayout.Y_AXIS));
        contentPanel.setBackground(new Color( 245,  245,  245));

        JLabel companyLabel = new JLabel( text: "<html><div style='text-align: center;'>For over 75
```

6. Evaluation

For evaluating our app, we conducted user studies involving participants from our target demographic, including homeowners and interior designers. The study aimed to gather feedback on the usability, functionality, and overall user experience of the Cozy Comfort Furniture app.

Methodology:

- Usability Testing: Participants were asked to perform specific tasks within the app, such as creating a new furniture design, adjusting colours, and saving designs. Their interactions and feedback were observed and recorded.
- Surveys: Participants completed surveys after using the app, providing quantitative feedback on various aspects such as ease of use, satisfaction with features, and overall impressions.
- Interviews: Follow-up interviews were conducted with participants to delve deeper into their experiences, preferences, and any challenges encountered while using the app.

Justification:

- Usability testing allowed us to observe firsthand how participants interacted with the app and identify any usability issues or pain points they encountered.
- Surveys provided quantitative data on user satisfaction and allowed us to gather feedback from a larger sample size quickly.
- Interviews provided qualitative insights into participants' experiences, allowing us to gain a deeper understanding of their needs and preferences.

Results:

Overall, the user studies revealed positive feedback regarding the app's ease of use, intuitive design, and helpful features such as the 2D and 3D visualization capabilities. However, some areas for improvement were identified,

- Enhancing the user interface to make certain features more discoverable and intuitive.
- Improving the performance and responsiveness of the app, particularly when rendering 3D designs.
- Providing more customization options and flexibility in adjusting furniture dimensions and styles.

Areas for Further Work and Improvement:

- Conducting usability testing with a larger and more diverse group of participants to ensure broader representation and gather a wider range of feedback.
- Iteratively testing and refining the app based on ongoing user feedback and usability studies.
- Exploring additional features and functionalities based on user needs and emerging trends in furniture design technology.

7. Summary

Cozy Comfort Furniture introduces a user-friendly desktop application designed to simplify furniture shopping by empowering users to effortlessly design and visualize furniture for their living spaces. With intuitive tools for visualization and design, the app allows users to create personalized layouts based on specific room dimensions, shapes, and colour schemes. Through methods such as surveys, interviews, and comparative analysis, user requirements were gathered and prioritized. User studies involving usability testing, surveys, and interviews revealed positive feedback on the app's ease of use, with areas for improvement identified. Overall, Cozy Comfort Furniture aims to provide a seamless and enjoyable experience for users seeking to transform their living spaces with custom furniture arrangements.

References

- Eckstein, R. M., 2003. *Java Swing*, s.l.: s.n.
- fakultet Sveu, u. f. S. u., 2009. *ZA PITANJA DRVNE TEHNOLOGIJE SCIENTIFIC AND PROFESSIONAL JOURNAL OF WOOD TECHNOLOGY IZDAVA^ I UREDNI[TVO Publisher and Editor's Office*, s.l.: s.n.
- Fjeld, M. M. o. E. a. E. E., 2006. *TableTop : First IEEE International Workshop on Horizontal Interactive Human-Computer Systems : Adelaide, Australia, 05-07 January 2006*, s.l.: s.n.
- Molina, J. P. G. P. G. P. M. F. M. F., n.d. *Bridging the gap: developing 2D and 3D user interfaces with the IDEAS methodology*, s.l.: s.n.
- Puder, A., n.d. *Extending Desktop Applications to the Web*, s.l.: s.n.

