

```

import javax.swing.*;
import javax.swing.border.AbstractBorder;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.RoundRectangle2D;
import java.awt.image.BufferedImage;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.*;

public class EcoEmailApp extends JFrame {
    private CardLayout cards = new CardLayout();
    private JPanel main = new JPanel(cards);
    private java.util.List<Rec> data = new ArrayList<>();
    private double co2, saved;
    private int emails;
    private JTextField user = new JTextField(15), num = new JTextField(10), size
= new JTextField(10);
    private JPasswordField pass = new JPasswordField(15);
    private JCheckBox c1 = new JCheckBox("Reduce attachments (-20%)"), c2 = new
JCheckBox("Green server (-10%)");
    private JLabel l1 = new JLabel("🌐 0kg"), l2 = new JLabel("🌿 0%"), l3 = new
JLabel("✉️ 0");
    private JTable t1 = new JTable(), t2 = new JTable();
    private JTextArea tips = new JTextArea();
    private JProgressBar bar = new JProgressBar(0, 100);
    private static final Color G = new Color(39, 174, 96), GL = new Color(46,
204, 113), B = new Color(236, 240, 241);
    private static final Font T = new Font("Segoe UI", Font.BOLD, 26), L = new
Font("Segoe UI", Font.PLAIN, 14), BF = new Font("Segoe UI", Font.BOLD, 13);

    static class Rec implements Serializable {
        String d; int e, s; double c, v; boolean ec;
        Rec(String d, int e, int s, double c, double v, boolean ec) { this.d = d;
this.e = e; this.s = s; this.c = c; this.v = v; this.ec = ec; }
    }

    class Btn extends JButton {
        Color bg, hv;
        Btn(String t, Color b, Color h) {
            super(t); bg = b; hv = h; setForeground(Color.WHITE); setFont(BF);

```

```

        setFocusPainted(false); setBorderPainted(false);
setContentAreaFilled(false); setCursor(new Cursor(Cursor.HAND_CURSOR));
        addMouseListener(new MouseAdapter() {
            public void mouseEntered(MouseEvent e) { setBackground(hv); }
            public void mouseExited(MouseEvent e) { setBackground(bg); }
        });
    }
    protected void paintComponent(Graphics g) {
        Graphics2D g2 = (Graphics2D) g.create();
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setColor(new Color(0, 0, 0, 30)); g2.fillRoundRect(3, 3,
getWidth() - 2, getHeight() - 2, 22, 22);
        g2.setColor(getBackground()); RoundRectangle2D r = new
RoundRectangle2D.Double(0, 0, getWidth(), getHeight(), 22, 22); g2.fill(r);
        g2.setColor(Color.WHITE); g2.setStroke(new BasicStroke(2));
g2.draw(r);
        FontMetrics fm = g2.getFontMetrics(getFont()); int x = (getWidth() -
fm.stringWidth(getText())) / 2, y = (getHeight() + fm.getAscent()) / 2;
        g2.setColor(getForeground()); g2.drawString(getText(), x, y);
g2.dispose();
    }
}

class Bdr extends AbstractBorder {
    int r = 12;
    public Insets getBorderInsets(Component c) { return new Insets(r + 4, r +
4, r + 4, r + 4); }
    public void paintBorder(Component c, Graphics g, int x, int y, int w, int
h) {
        Graphics2D g2 = (Graphics2D) g;
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setColor(GL); g2.setStroke(new BasicStroke(2.5f));
g2.drawRoundRect(x + 1, y + 1, w - 3, h - 3, r * 2, r * 2);
    }
}

public EcoEmailApp() {
    try
{ UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName()); } catch
(Exception e1) {

```

```

        try
    { UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName()); }
    catch (Exception e2) {} }

    setTitle("🌿 Eco Email Dashboard"); setSize(1100, 750);
    setLocationRelativeTo(null); setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
    setIconImage(icon()); add(main); cards.show(main, "login");
    addWindowListener(new WindowAdapter() { public void
    windowClosing(WindowEvent e) { save(); System.exit(0); } });
    load(); screens(); setVisible(true);
    }

    // ENHANCED: App icon with leaf design
    private Image icon() {
        BufferedImage i = new BufferedImage(48, 48, BufferedImage.TYPE_INT_ARGB);
        Graphics2D g = i.createGraphics();
        g.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);
        g.setColor(G); g.fillOval(0, 0, 48, 48);
        g.setColor(GL); g.fillOval(8, 8, 32, 32);
        g.setColor(Color.WHITE); g.fillOval(16, 16, 16, 16);
        g.setColor(G); g.fillRoundRect(18, 10, 12, 28, 10, 10); // Leaf shape
        g.dispose(); return i;
    }

    // NEW: Custom icon creator for labels
    private ImageIcon createIcon(String type, int size) {
        BufferedImage img = new BufferedImage(size, size,
        BufferedImage.TYPE_INT_ARGB);
        Graphics2D g = img.createGraphics();
        g.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);

        switch(type) {
            case "user":
                // User icon - person silhouette
                g.setColor(G); g.fillOval(size/3, size/6, size/3, size/3); //
                Head
                g.fillRoundRect(size/4, size/2, size/2, size/2, 20, 20); // Body
                break;
            case "lock":
                // Lock icon
                g.setColor(G); g.drawRoundRect(size/4, size/3, size/2, size/2,
                10, 10); // Body

```

```

        g.setStroke(new BasicStroke(3)); g.drawArc(size/3, size/8,
size/3, size/2, 0, 180); // Shackle
        g.setColor(GL); g.fillOval(size/2 - 4, size/2, 8, 8); // Keyhole
        break;
    case "email":
        // Email envelope icon
        g.setColor(GL); g.fillRoundRect(size/6, size/4, 2*size/3, size/2,
8, 8);

        g.setColor(G); int[] xp = {size/6, size/2, 5*size/6}; int[] yp =
{size/4, size/2, size/4};
        g.fillPolygon(xp, yp, 3);
        break;
    case "calculator":
        // Calculator icon
        g.setColor(G); g.fillRoundRect(size/4, size/8, size/2, 3*size/4,
8, 8);

        g.setColor(Color.WHITE);
        for(int r = 0; r < 3; r++) for(int c = 0; c < 2; c++)
            g.fillRect(size/4 + 8 + c*12, size/8 + 12 + r*12, 8, 8);
        break;
    case "earth":
        // Earth/globe icon
        g.setColor(new Color(33, 150, 243)); g.fillOval(0, 0, size,
size);

        g.setColor(GL); g.fillOval(size/6, size/6, 2*size/3, 2*size/3);
        g.setColor(G);
        g.fillOval(size/4, size/3, size/5, size/5);
        g.fillOval(size/2, size/4, size/6, size/6);
        break;
    case "leaf":
        // Leaf icon
        g.setColor(G); g.fillRoundRect(size/4, size/8, size/2, 3*size/4,
size/3, size/3);

        g.setColor(GL); g.fillOval(size/3, size/5, size/4, size/4);
        g.setColor(new Color(139, 195, 74));
        g.setStroke(new BasicStroke(2));
        g.drawLine(size/2, size/4, size/2, 3*size/4);
        break;
    case "chart":
        // Bar chart icon
        g.setColor(G);
        g.fillRect(size/6, 2*size/3, size/8, size/4);
        g.fillRect(size/3, size/2, size/8, size/2);
        g.fillRect(size/2, size/3, size/8, 2*size/3);

```

```

        g.fillRect(2*size/3, size/4, size/8, 3*size/4);
        break;
    case "settings":
        // Gear/settings icon
        g.setColor(G); g.fillOval(size/3, size/3, size/3, size/3);
        g.setColor(GL);
        for(int i = 0; i < 8; i++) {
            double angle = i * Math.PI / 4;
            int x = (int)(size/2 + size/3 * Math.cos(angle));
            int y = (int)(size/2 + size/3 * Math.sin(angle));
            g.fillRect(x-2, y-2, 4, 8);
        }
        break;
    }
    g.dispose();
    return new ImageIcon(img);
}

private void screens() { main.add(login(), "login"); main.add(entry(),
"entry"); main.add(dash(), "dash"); main.add(reports(), "reports");
main.add(admin(), "admin"); }

private JPanel login() {
    JPanel p = new JPanel(new BorderLayout());
    JPanel bg = new JPanel() {
        protected void paintComponent(Graphics g) { super.paintComponent(g);
Graphics2D g2 = (Graphics2D) g;
            g2.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);
            GradientPaint gp = new GradientPaint(0, 0, GL, 0, getHeight(),
G); g2.setPaint(gp); g2.fillRect(0, 0, getWidth(), getHeight()); }
    }; bg.setLayout(new GridBagLayout());

    JPanel card = new JPanel(); card.setLayout(new BoxLayout(card,
BoxLayout.Y_AXIS));
    card.setBackground(Color.WHITE);
    card.setBorder(BorderFactory.createCompoundBorder(new Bdr(),
BorderFactory.createEmptyBorder(40, 50, 40, 50)));
    card.setPreferredSize(new Dimension(500, 480));

    // ENHANCED: Title with leaf icon
    JPanel titlePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    titlePanel.setBackground(Color.WHITE);

```

```

        JLabel leafIcon = new JLabel(createIcon("leaf", 48));
        titlePanel.add(leafIcon);
        JLabel title = new JLabel("Eco Email Tracker"); title.setFont(T);
title.setForeground(G.darker());
        titlePanel.add(title);
        titlePanel.setAlignmentX(Component.CENTER_ALIGNMENT);
        titlePanel.setBorder(BorderFactory.createEmptyBorder(20, 0, 10, 0));
        card.add(titlePanel);

        JLabel subtitle = new JLabel("Track your email carbon footprint");
subtitle.setFont(new Font("Segoe UI", Font.ITALIC, 15));
        subtitle.setForeground(new Color(100, 100, 100));
subtitle.setAlignmentX(Component.CENTER_ALIGNMENT);
        subtitle.setBorder(BorderFactory.createEmptyBorder(0, 0, 30, 0));
card.add(subtitle);

        // ENHANCED: Username with icon
JPanel userPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 5, 0));
userPanel.setBackground(Color.WHITE);
userPanel.add(new JLabel(createIcon("user", 20)));
        JLabel userLabel = new JLabel("Username:"); userLabel.setFont(L);
userLabel.setForeground(G.darker());
        userPanel.add(userLabel);
        userPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
        userPanel.setBorder(BorderFactory.createEmptyBorder(12, 0, 6, 0));
        card.add(userPanel);

        user.setFont(L); user.setBorder(new Bdr());
user.setBackground(Color.WHITE);
        user.setMaximumSize(new Dimension(320, 42));
user.setAlignmentX(Component.CENTER_ALIGNMENT); card.add(user);

        // ENHANCED: Password with icon
JPanel passPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 5, 0));
passPanel.setBackground(Color.WHITE);
passPanel.add(new JLabel(createIcon("lock", 20)));
        JLabel passLabel = new JLabel("Password:"); passLabel.setFont(L);
passLabel.setForeground(G.darker());
        passPanel.add(passLabel);
        passPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
        passPanel.setBorder(BorderFactory.createEmptyBorder(18, 0, 6, 0));
        card.add(passPanel);

```

```

        pass.setFont(L); pass.setBorder(new Bdr());
pass.setBackground(Color.WHITE);
        pass.setMaximumSize(new Dimension(320, 42));
pass.setAlignmentX(Component.CENTER_ALIGNMENT); card.add(pass);

        card.add(Box.createVerticalStrut(28));

        // ENHANCED: Login button with Leaf icon
JPanel btnPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        btnPanel.setBackground(Color.WHITE);
        btnPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
        Btn loginBtn = new Btn("🍃 Login to Dashboard", G, GL);
loginBtn.setPreferredSize(new Dimension(240, 48));
        loginBtn.addActionListener(e -> {
            String u = user.getText().trim(), pw = new
String(pass.getPassword()).trim();
            if (u.isEmpty() || pw.isEmpty())
{ JOptionPane.showMessageDialog(this, "Please enter both username and password",
"Login Required", JOptionPane.WARNING_MESSAGE); return; }
            user.setText(""); pass.setText("");
            cards.show(main, u.equals("admin") && pw.equals("admin") ? "admin" :
"dash"); updateDash();
        });
        btnPanel.add(loginBtn);
        card.add(btnPanel);

        bg.add(card); p.add(bg, BorderLayout.CENTER); return p;
    }

    private JPanel entry() {
        JPanel p = new JPanel(new BorderLayout());
        JPanel bg = new JPanel() {
            protected void paintComponent(Graphics g) { super.paintComponent(g);
Graphics2D g2 = (Graphics2D) g;
                g2.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);
                GradientPaint gp = new GradientPaint(0, 0, new Color(52, 152,
219), 0, getHeight(), GL); g2.setPaint(gp); g2.fillRect(0, 0, getWidth(),
getHeight()); }
        }; bg.setLayout(new GridBagLayout());

        JPanel card = new JPanel(); card.setLayout(new BoxLayout(card,
BoxLayout.Y_AXIS));

```

```

        card.setBackground(Color.WHITE);
card.setBorder(BorderFactory.createCompoundBorder(new Bdr(),
BorderFactory.createEmptyBorder(35, 45, 35, 45)));
        card.setPreferredSize(new Dimension(500, 540));

        // ENHANCED: Title with calculator icon
JPanel titlePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
titlePanel.setBackground(Color.WHITE);
titlePanel.add(new JLabel(createIcon("calculator", 42)));
JLabel title = new JLabel(" Email Carbon Calculator"); title.setFont(T);
title.setForeground(G.darker());
titlePanel.add(title);
titlePanel.setAlignmentX(Component.CENTER_ALIGNMENT);
titlePanel.setBorder(BorderFactory.createEmptyBorder(18, 0, 25, 0));
card.add(titlePanel);

        // ENHANCED: Email count with icon
JPanel numPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 5, 0));
numPanel.setBackground(Color.WHITE);
numPanel.add(new JLabel(createIcon("email", 20)));
JLabel numLabel = new JLabel("Number of Emails:"); numLabel.setFont(L);
numLabel.setForeground(G.darker());
numPanel.add(numLabel);
numPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
numPanel.setBorder(BorderFactory.createEmptyBorder(8, 0, 6, 0));
card.add(numPanel);

        num.setFont(L); num.setBorder(new Bdr()); num.setMaximumSize(new
Dimension(300, 40)); num.setAlignmentX(Component.CENTER_ALIGNMENT);
card.add(num);

        // ENHANCED: Size with chart icon
JPanel sizePanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 5, 0));
sizePanel.setBackground(Color.WHITE);
sizePanel.add(new JLabel(createIcon("chart", 20)));
JLabel sizeLabel = new JLabel("Average Size per Email (KB):");
sizeLabel.setFont(L); sizeLabel.setForeground(G.darker());
sizePanel.add(sizeLabel);
sizePanel.setAlignmentX(Component.CENTER_ALIGNMENT);
sizePanel.setBorder(BorderFactory.createEmptyBorder(16, 0, 6, 0));
card.add(sizePanel);

```





```

        g2.drawLine(x+4, y+9, x+7, y+13);
        g2.drawLine(x+7, y+13, x+14, y+5);
    }
}

public int getIconWidth() { return 18; }
public int getIconHeight() { return 18; }
};
}

private void calc() {
    try {
        int e = Integer.parseInt(num.getText().trim()), s =
Integer.parseInt(size.getText().trim());
        if (e <= 0 || s < 0) { JOptionPane.showMessageDialog(this, "Enter
valid numbers (emails > 0, size ≥ 0)", "Invalid Input",
JOptionPane.ERROR_MESSAGE); return; }
        double base = (e * 0.004) + (e * s * 0.000019), red =
(c1.isSelected() ? 0.2 : 0) + (c2.isSelected() ? 0.1 : 0);
        double c = base * (1 - red), v = base * red; co2 += c; saved += v;
emails += e;
        String d = new SimpleDateFormat("dd/MM").format(new Date());
data.add(new Rec(d, e, s, c, v, red > 0));
        num.setText(""); size.setText(""); c1.setSelected(false);
c2.setSelected(false);

        // ENHANCED: Success dialog with custom icon
        JLabel msgLabel = new JLabel(String.format("<html><center><h2>☑
Calculation Complete!</h2><p>📧 <b>%d emails</b> processed</p><p>🌍 Emissions:
<b>%.3f kg CO2</b></p><p>💚 Saved: <b>%.3f kg CO2</b></p></center></html>", e,
c, v));
        msgLabel.setIcon(createIcon("earth", 64));
        JOptionPane.showMessageDialog(this, msgLabel, "Success",
JOptionPane.PLAIN_MESSAGE);

        save(); cards.show(main, "dash"); updateDash();
    } catch (NumberFormatException ex) { JOptionPane.showMessageDialog(this,
"Please enter valid numbers only", "Input Error", JOptionPane.ERROR_MESSAGE); }
}

private JPanel dash() {
    JPanel p = new JPanel(new BorderLayout());
    JPanel bg = new JPanel() {

```

```

        protected void paintComponent(Graphics g) { super.paintComponent(g);
Graphics2D g2 = (Graphics2D) g;
            g2.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);
                GradientPaint gp = new GradientPaint(0, 0, G, 0, getHeight(), new
Color(29, 105, 20)); g2.setPaint(gp); g2.fillRect(0, 0, getWidth(),
getHeight()); }
        }; bg.setLayout(new BorderLayout());

        JPanel card = new JPanel(new BorderLayout()); card.setBackground(B);
        card.setBorder(BorderFactory.createCompoundBorder(new Bdr(),
BorderFactory.createEmptyBorder(25, 25, 25, 25)));
        card.setPreferredSize(new Dimension(920, 620));

        JLabel title = new JLabel("🏡 Eco Dashboard", JLabel.CENTER);
title.setFont(T); title.setForeground(G.darker());
        title.setBorder(BorderFactory.createEmptyBorder(15, 0, 15, 0));
        JPanel nav = new JPanel(new FlowLayout(FlowLayout.RIGHT));
nav.setBackground(B);
        Btn entryBtn = new Btn("➕ New Entry", G, GL);
entryBtn.setPreferredSize(new Dimension(110, 32)); entryBtn.addActionListener(e -
> cards.show(main, "entry")); nav.add(entryBtn);
        Btn rptBtn = new Btn("📄 Reports", GL, G); rptBtn.setPreferredSize(new
Dimension(95, 32)); rptBtn.addActionListener(e -> { cards.show(main, "reports");
updateRpt(); }); nav.add(rptBtn);
        Btn logoutBtn = new Btn("🚪 Logout", Color.ORANGE, Color.RED);
logoutBtn.setPreferredSize(new Dimension(85, 32)); logoutBtn.addActionListener(e
-> { save(); cards.show(main, "login"); }); nav.add(logoutBtn);
        JPanel top = new JPanel(new BorderLayout()); top.setBackground(B);
top.add(title, BorderLayout.CENTER); top.add(nav, BorderLayout.EAST);
card.add(top, BorderLayout.NORTH);

        JPanel content = new JPanel(new GridBagLayout());
content.setBackground(B);
        GridBagConstraints gbc = new GridBagConstraints(); gbc.insets = new
Insets(15, 15, 15, 15); gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 3; gbc.fill
= GridBagConstraints.HORIZONTAL;
        JPanel stats = new JPanel(new GridLayout(1, 3, 15, 0));
stats.setBackground(B);

        // ENHANCED: Stat cards with custom icons

```

```

        JPanel c1 = new JPanel(new BorderLayout());
c1.setBackground(Color.WHITE); c1.setBorder(new Bdr()); c1.setPreferredSize(new
Dimension(260, 100));
        JLabel ic1 = new JLabel(createIcon("earth", 36), JLabel.CENTER);
ic1.setBorder(BorderFactory.createEmptyBorder(8, 0, 4, 0)); c1.add(ic1,
BorderLayout.NORTH);
        l1.setFont(new Font("Segoe UI", Font.BOLD, 16));
l1.setHorizontalAlignment(SwingConstants.CENTER); l1.setForeground(G); c1.add(l1,
BorderLayout.CENTER); stats.add(c1);

        JPanel c2 = new JPanel(new BorderLayout());
c2.setBackground(Color.WHITE); c2.setBorder(new Bdr()); c2.setPreferredSize(new
Dimension(260, 100));
        JLabel ic2 = new JLabel(createIcon("leaf", 36), JLabel.CENTER);
ic2.setBorder(BorderFactory.createEmptyBorder(8, 0, 4, 0)); c2.add(ic2,
BorderLayout.NORTH);
        l2.setFont(new Font("Segoe UI", Font.BOLD, 16));
l2.setHorizontalAlignment(SwingConstants.CENTER); l2.setForeground(G); c2.add(l2,
BorderLayout.CENTER); stats.add(c2);

        JPanel c3 = new JPanel(new BorderLayout());
c3.setBackground(Color.WHITE); c3.setBorder(new Bdr()); c3.setPreferredSize(new
Dimension(260, 100));
        JLabel ic3 = new JLabel(createIcon("email", 36), JLabel.CENTER);
ic3.setBorder(BorderFactory.createEmptyBorder(8, 0, 4, 0)); c3.add(ic3,
BorderLayout.NORTH);
        l3.setFont(new Font("Segoe UI", Font.BOLD, 16));
l3.setHorizontalAlignment(SwingConstants.CENTER); l3.setForeground(G); c3.add(l3,
BorderLayout.CENTER); stats.add(c3);

        content.add(stats, gbc);

        gbc.gridy = 1; bar.setPreferredSize(new Dimension(800, 28));
bar.setBackground(B); bar.setForeground(G); bar.setBorder(new Bdr());
content.add(bar, gbc);
        gbc.gridy = 2; gbc.gridwidth = 1; gbc.weightx = 0.55; gbc.weighty = 1;
gbc.fill = GridBagConstraints.BOTH;
        t1.setFont(L); t1.setRowHeight(28); t1.setBackground(Color.WHITE);
JScrollPane sp1 = new JScrollPane(t1); sp1.setPreferredSize(new Dimension(480,
200)); sp1.setBorder(new Bdr()); content.add(sp1, gbc);
        gbc.gridx = 1; gbc.weightx = 0.45; tips.setFont(new Font("Segoe UI",
Font.PLAIN, 13)); tips.setLineWrap(true); tips.setWrapStyleWord(true);
tips.setEditable(false); tips.setBackground(Color.WHITE);

```

```

        JScrollPane sp2 = new JScrollPane(tips); sp2.setPreferredSize(new
Dimension(360, 200)); sp2.setBorder(new Bdr()); content.add(sp2, gbc);
        card.add(content, BorderLayout.CENTER);

        JPanel center = new JPanel(new GridBagLayout());
center.setBackground(Color.WHITE); GridBagConstraints cgbc = new
GridBagConstraints(); cgbc.insets = new Insets(35, 0, 35, 0);
        center.add(card, cgbc); bg.add(center, BorderLayout.CENTER); p.add(bg,
BorderLayout.CENTER); return p;
    }

    private void updateDash() {
        if (data.isEmpty()) { l1.setText("🌐 0 kg"); l2.setText("💧 0%");
l3.setText("✉️ 0"); bar.setValue(0);
            tips.setText("👋 Welcome to Eco Email Tracker!\n\nStart by clicking
' + New Entry' to calculate your first email carbon footprint.\n\n💡 Enable eco
options to reduce emissions by up to 30%!");
            t1.setModel(new DefaultTableModel(new Object[0][0], new
String[]{"Metric", "Value"})); return; }
            double score = Math.max(0, Math.min(100, 100 - co2 * 800));
l1.setText(String.format("🌐 %.3f kg", co2));
l2.setText(String.format("💧 %.0f%%", score)); l3.setText(String.format("✉️ %d",
emails));
            bar.setValue((int) score); bar.setString(String.format("Eco
Score: %.0f%%", score)); bar.setStringPainted(true);
            t1.setModel(new DefaultTableModel(new Object[][]{{"Total Emails",
emails}, {"Avg CO2/Email", String.format("%.4f kg", co2 / data.size())},
{"Total Emissions", String.format("%.3f kg", co2)}, {"Total Saved",
String.format("%.3f kg", saved)}, {"Eco Score", String.format("%.0f%%", score)}}},
new String[]{"Metric", "Value"}));
            tips.setText(String.format("📊 Your Eco Progress:\n\n• %d emails
tracked\n• %.3f kg total CO2 emissions\n• %.3f kg saved with eco options\n•
Current eco score: %.0f%%\n\n💡 Quick Tips:\n" +
                "• Compress attachments before sending\n• Use eco-friendly email
providers\n• Enable green server options\n• Batch similar messages together",
emails, co2, saved, score));
    }

    private JPanel reports() {
        JPanel p = new JPanel(new BorderLayout());
        JPanel bg = new JPanel() {

```

```

        protected void paintComponent(Graphics g) { super.paintComponent(g);
Graphics2D g2 = (Graphics2D) g;
        g2.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);
        GradientPaint gp = new GradientPaint(0, 0, new Color(139, 195,
74), 0, getHeight(), GL); g2.setPaint(gp); g2.fillRect(0, 0, getWidth(),
getHeight()); }
    }; bg.setLayout(new BorderLayout());

    JPanel card = new JPanel(new BorderLayout()); card.setBackground(B);
    card.setBorder(BorderFactory.createCompoundBorder(new Bdr(),
BorderFactory.createEmptyBorder(25, 25, 25, 25)));
    card.setPreferredSize(new Dimension(880, 570));

    JPanel top = new JPanel(new BorderLayout()); top.setBackground(B);
    // ENHANCED: Title with chart icon
    JPanel titlePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    titlePanel.setBackground(B);
    titlePanel.add(new JLabel(createIcon("chart", 32)));
    JLabel title = new JLabel(" Detailed Reports"); title.setFont(T);
title.setForeground(G.darker());
    titlePanel.add(title);
    top.add(titlePanel, BorderLayout.CENTER);

    JPanel nav = new JPanel(new FlowLayout(FlowLayout.RIGHT));
nav.setBackground(B);
    Btn backBtn = new Btn("⬅ Back to Dashboard", G, GL);
backBtn.setPreferredSize(new Dimension(160, 32)); backBtn.addActionListener(e ->
cards.show(main, "dash")); nav.add(backBtn);
    top.add(nav, BorderLayout.EAST); card.add(top, BorderLayout.NORTH);

    t2.setFont(L); t2.setRowHeight(28); t2.setBackground(Color.WHITE);
JScrollPane scroll = new JScrollPane(t2);
    scroll.setPreferredSize(new Dimension(830, 470)); scroll.setBorder(new
Bdr()); card.add(scroll, BorderLayout.CENTER);

    JPanel center = new JPanel(new GridBagLayout());
center.setBackground(Color.WHITE); GridBagConstraints cgbc = new
GridBagConstraints();
    cgbc.insets = new Insets(40, 0, 40, 0); center.add(card, cgbc);
bg.add(center, BorderLayout.CENTER); p.add(bg, BorderLayout.CENTER); return p;
}

```

```

private void updateRpt() {
    if (data.isEmpty()) { t2.setModel(new DefaultTableModel(new Object[0][0],
new String[]{"Date", "Emails", "Size", "CO2", "Saved", "Eco"})); return; }
    DefaultTableModel m = new DefaultTableModel(new String[]{"Date",
"Emails", "Size (KB)", "CO2 (kg)", "Saved (kg)", "Eco Options"}, 0);
    for (Rec r : data) m.addRow(new Object[]{r.d, r.e, r.s,
String.format("%.3f", r.c), String.format("%.3f", r.v), r.ec ? "☑ Yes" : "✗
No"}); t2.setModel(m);
}

private JPanel admin() {
    JPanel p = new JPanel(new BorderLayout());
    JPanel bg = new JPanel() {
        protected void paintComponent(Graphics g) { super.paintComponent(g);
Graphics2D g2 = (Graphics2D) g;
            g2.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);
            GradientPaint gp = new GradientPaint(0, 0, new Color(156, 39,
176), 0, getHeight(), new Color(186, 104, 200)); g2.setPaint(gp); g2.fillRect(0,
0, getWidth(), getHeight()); }
    }; bg.setLayout(new BorderLayout());

    JPanel card = new JPanel(new BorderLayout()); card.setBackground(B);
    card.setBorder(BorderFactory.createCompoundBorder(new Bdr(),
BorderFactory.createEmptyBorder(30, 30, 30, 30)));
    card.setPreferredSize(new Dimension(620, 470));

    JPanel top = new JPanel(new BorderLayout()); top.setBackground(B);
    // ENHANCED: Title with settings icon
    JPanel titlePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    titlePanel.setBackground(B);
    titlePanel.add(new JLabel(createIcon("settings", 32)));
    JLabel title = new JLabel(" Admin Control Panel"); title.setFont(T);
title.setForeground(new Color(156, 39, 176));
    titlePanel.add(title);
    top.add(titlePanel, BorderLayout.CENTER);

    JPanel nav = new JPanel(new FlowLayout(FlowLayout.RIGHT));
nav.setBackground(B);
    Btn logoutBtn = new Btn("🔒 Logout", Color.ORANGE, Color.RED);
logoutBtn.setPreferredSize(new Dimension(90, 32)); logoutBtn.addActionListener(e
-> { save(); cards.show(main, "login"); }); nav.add(logoutBtn);
    top.add(nav, BorderLayout.EAST); card.add(top, BorderLayout.NORTH);

```



```

        JPanel btns = new JPanel(new GridLayout(3, 2, 18, 18));
btns.setBackground(B); btns.setBorder(BorderFactory.createEmptyBorder(35, 45, 45,
45));

        String[] acts = {"👤 Manage Users", "📊 System Analytics", "⚙️
Configuration", "📄 Generate Reports", "💾 Backup Database", "🔄 Reset
Settings"};

        Color[] cols = {G, GL, new Color(33, 150, 243), new Color(255, 193, 7),
new Color(156, 39, 176), Color.GRAY};
        for (int i = 0; i < acts.length; i++) { int idx = i; Btn btn = new
Btn(acts[i], cols[i], cols[i].brighter()); btn.setPreferredSize(new
Dimension(230, 54));

            btn.addActionListener(e -> JOptionPane.showMessageDialog(this, "Admin
function: " + acts[idx] + "\n\nThis feature would manage:\n• User permissions\n•
System settings\n• Data exports", "Admin Panel",
JOptionPane.INFORMATION_MESSAGE)); btns.add(btn); }
        card.add(btns, BorderLayout.CENTER);

        JPanel center = new JPanel(new GridBagLayout());
center.setBackground(Color.WHITE); GridBagConstraints cgbc = new
GridBagConstraints();
        cgbc.insets = new Insets(75, 0, 75, 0); center.add(card, cgbc);
bg.add(center, BorderLayout.CENTER); p.add(bg, BorderLayout.CENTER); return p;
    }

    private void save() {
        try (ObjectOutputStream o = new ObjectOutputStream(new
FileOutputStream("eco_data.dat"))) {
            Map<String, Object> d = new HashMap<>(); d.put("data", data);
d.put("co2", co2); d.put("saved", saved); d.put("emails", emails);
o.writeObject(d);

            System.out.println("✅ Saved: " + data.size() + " records");
        } catch (IOException e) { System.err.println("❌ Save error: " +
e.getMessage()); }
    }

    @SuppressWarnings("unchecked")
    private void load() {
        File f = new File("eco_data.dat");
        if (!f.exists()) { data.add(new Rec("14/10", 50, 120, 0.15, 0.03, true));
co2 = 0.15; saved = 0.03; emails = 50; save(); return; }
        try (ObjectInputStream o = new ObjectInputStream(new FileInputStream(f)))
    {

```



```

        Map<String, Object> d = (Map<String, Object>) o.readObject(); data =
(java.util.List<Rec>) d.get("data");
        co2 = (Double) d.getDefault("co2", 0.0); saved = (Double)
d.getDefault("saved", 0.0); emails = (Integer) d.getDefault("emails", 0);
        System.out.println("☑ Loaded: " + data.size() + " records");
    } catch (Exception e) { System.err.println("✗ Load error: " +
e.getMessage()); data.clear(); co2 = 0; saved = 0; emails = 0; }
    }

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

```