

TaxHelper Project Structure

This markdown file contains the actual source code and structure for the TaxHelper Java application.

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        new ui.IntroAnimation();  
    }  
}
```

ui/

IntroAnimation.java

```
package ui;  
  
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.Font;  
import java.awt.Image;  
import javax.swing.BorderFactory;  
import javax.swing.ImageIcon;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.Timer;  
  
public class IntroAnimation extends JFrame {  
    public IntroAnimation() {  
        setUndecorated(true); // remove window borders  
        setSize(400, 300);  
        setLocationRelativeTo(null);  
        setLayout(new BorderLayout());  
        getContentPane().setBackground(Color.WHITE);  
  
        // ----- LOGO CENTER -----  
        ImageIcon logoIcon = new ImageIcon("ui/images/logo.png");  
        Image scaledImage = logoIcon.getImage().getScaledInstance(120, 120,  
Image.SCALE_SMOOTH);  
        JLabel logoLabel = new JLabel(new ImageIcon(scaledImage));  
        logoLabel.setHorizontalAlignment(JLabel.CENTER);  
        logoLabel.setVerticalAlignment(JLabel.CENTER);  
        add(logoLabel, BorderLayout.CENTER);  
  
        // ----- TEXT BOTTOM -----  
        JLabel textLabel = new JLabel("TaxHelper App");
```

```

        textLabel.setFont(new Font("Segoe UI", Font.BOLD, 20));
        textLabel.setForeground(new Color(50, 50, 50));
        textLabel.setHorizontalAlignment(JLabel.CENTER);
        textLabel.setBorder(BorderFactory.createEmptyBorder(10, 0, 20, 0));
        add(textLabel, BorderLayout.SOUTH);

        setVisible(true);

        // Delay + open main frame in background
        Timer timer = new Timer(2500, e -> {
            ((Timer) e.getSource()).stop(); // Important!
            dispose();                       // Close splash screen
            new MainFrame();                 // Open main app window
        });
        timer.setRepeats(false);
        timer.start();
    }
}

```

MainFrame.java

```

package ui;

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

public class MainFrame extends JFrame implements ActionListener {

    JRadioButton rbFullTime;
    JRadioButton rbPartTime;
    JRadioButton rbFreelance;
    JButton button;
    ButtonGroup jobGroup;

    public MainFrame() {

        // Frame settings
        setTitle("TaxHelper App");
        setSize(500, 400);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setIconImage(new ImageIcon("ui/images/logo.png").getImage());

        setLayout(new BorderLayout());
        getContentPane().setBackground(Color.WHITE);

        // ----- HEADER -----
    }
}

```

```
JPanel headerPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15,
20));
headerPanel.setBackground(new Color(245, 245, 245));

ImageIcon rawIcon = new ImageIcon("ui/images/job.png");
Image scaledImg = rawIcon.getImage().getScaledInstance(30, 30,
Image.SCALE_SMOOTH);
JLabel iconLabel = new JLabel(new ImageIcon(scaledImg));

JLabel titleLabel = new JLabel("Select your job");
titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 22));
titleLabel.setForeground(new Color(30, 30, 30));

headerPanel.add(iconLabel);
headerPanel.add(titleLabel);
add(headerPanel, BorderLayout.NORTH);

// ----- RADIO BUTTONS -----
JPanel radioPanel = new JPanel();
radioPanel.setLayout(new BoxLayout(radioPanel, BoxLayout.Y_AXIS));
radioPanel.setBackground(Color.WHITE);
radioPanel.setBorder(BorderFactory.createEmptyBorder(30, 70, 30, 70));

rbFullTime = new JRadioButton("Full-time");
rbPartTime = new JRadioButton("Part-time");
rbFreelance = new JRadioButton("Freelance");

for (JRadioButton rb : new JRadioButton[]{rbFullTime, rbPartTime,
rbFreelance}) {
    rb.setFont(new Font("Consolas", Font.PLAIN, 18));
    rb.setFocusable(false);
    rb.setBackground(Color.WHITE);
    rb.setAlignmentX(Component.LEFT_ALIGNMENT);
    radioPanel.add(rb);
    radioPanel.add(Box.createRigidArea(new Dimension(0, 10)));
}

jobGroup = new ButtonGroup();
jobGroup.add(rbFullTime);
jobGroup.add(rbPartTime);
jobGroup.add(rbFreelance);

add(radioPanel, BorderLayout.CENTER);

// ----- BUTTON -----
JPanel buttonPanel = new JPanel();
buttonPanel.setBackground(Color.WHITE);

button = new JButton("Submit");
button.setFont(new Font("Segoe UI", Font.BOLD, 16));
button.setBackground(new Color(72, 201, 111));
button.setForeground(Color.WHITE);
button.setFocusPainted(false);           // removes blue ring when clicked
button.setBorderPainted(false);          // removes visible border
```

```

        button.setContentAreaFilled(true);           // keeps the background fill
        button.setOpaque(true);                     // ensures solid background

        button.setPreferredSize(new Dimension(140, 40));
        button.addActionListener(this);

        buttonPanel.add(button);
        add(buttonPanel, BorderLayout.SOUTH);

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (rbFullTime.isSelected()) {
            JOptionPane.showMessageDialog(this, "You selected: Full-time",
                "Selection", JOptionPane.INFORMATION_MESSAGE);
            Database.set("job", "Full-time");
            new IncomeFrame();
            this.dispose();
        } else if (rbPartTime.isSelected()) {
            JOptionPane.showMessageDialog(this, "You selected: Part-time",
                "Selection", JOptionPane.INFORMATION_MESSAGE);
            Database.set("job", "Part-time");
            new IncomeFrame();
            this.dispose();
        } else if (rbFreelance.isSelected()) {
            JOptionPane.showMessageDialog(this, "You selected: Freelance",
                "Selection", JOptionPane.INFORMATION_MESSAGE);
            Database.set("job", "Freelance");
            new IncomeFrame();
            this.dispose();
        } else {
            JOptionPane.showMessageDialog(this, "Please select a job before
                submitting.", "Warning", JOptionPane.WARNING_MESSAGE);
        }
    }
}

```

IncomeFrame.java

```

package ui;

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

```

```

public class IncomeFrame extends JFrame implements ActionListener {

    JTextField incomeField;
    JButton submitButton;

    public IncomeFrame() {
        setTitle("TaxHelper - Income Input");
        setSize(500, 250);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE); // only closes this
window
        setIconImage(new ImageIcon("ui/images/logo.png").getImage());
        setLayout(new BorderLayout(10, 10));
        getContentPane().setBackground(Color.WHITE);

        // ----- HEADER -----
        JLabel topLabel = new JLabel("Insert your income here");
        topLabel.setFont(new Font("Segoe UI", Font.BOLD, 20));
        topLabel.setHorizontalAlignment(JLabel.CENTER);
        topLabel.setBorder(BorderFactory.createEmptyBorder(20, 0, 0, 0));
        add(topLabel, BorderLayout.NORTH);

        // ----- CENTER FIELD + BUTTON -----
        JPanel centerPanel = new JPanel();
        centerPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 40));
        centerPanel.setBackground(Color.WHITE);

        incomeField = new JTextField(20);
        incomeField.setFont(new Font("Consolas", Font.PLAIN, 18));

        submitButton = new JButton("Submit");
        submitButton.setFont(new Font("Segoe UI", Font.BOLD, 16));
        submitButton.setBackground(new Color(72, 201, 111));
        submitButton.setForeground(Color.WHITE);
        submitButton.setFocusPainted(false); // removes blue ring when
clicked
        submitButton.setBorderPainted(false); // removes visible border
        submitButton.setContentAreaFilled(true); // keeps the background
fill
        submitButton.setOpaque(true); // ensures solid background
        submitButton.addActionListener(this);

        centerPanel.add(incomeField);
        centerPanel.add(submitButton);

        add(centerPanel, BorderLayout.CENTER);

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == submitButton) {
            // Optional: You can do something with the input first

```

```

        String input = incomeField.getText().trim();
        System.out.println("User entered: " + input);
        double income = Double.parseDouble(input);
        Database.set("income", income);

        // Then clear the text
        incomeField.setText("");
        new DeductionFrame();
        this.dispose();
    }
}

```

DeductionFrame.java

```

package ui;

import Services.Database;
import Services.TaxCalculator;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class DeductionFrame extends JFrame implements ActionListener {

    JCheckBox cbEducation;
    JCheckBox cbHealth;
    JCheckBox cbDonation;
    JButton submitButton;

    public DeductionFrame() {
        setTitle("TaxHelper - Deduction Selection");
        setSize(500, 400);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setIconImage(new ImageIcon("ui/images/logo.png").getImage());
        getContentPane().setBackground(Color.WHITE);
        setLayout(new BorderLayout());

        // ----- HEADER -----
        JPanel headerPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15,
20));
        headerPanel.setBackground(new Color(245, 245, 245));

        ImageIcon rawIcon = new ImageIcon("ui/images/deduction.png");
    }
}

```

```

        Image scaledImg = rawIcon.getImage().getScaledInstance(30, 30,
Image.SCALE_SMOOTH);
        JLabel iconLabel = new JLabel(new ImageIcon(scaledImg));

        JLabel titleLabel = new JLabel("Select your deductions");
        titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 22));
        titleLabel.setForeground(new Color(30, 30, 30));

        headerPanel.add(iconLabel);
        headerPanel.add(titleLabel);

        add(headerPanel, BorderLayout.NORTH);

        // ----- CHECKBOX PANEL -----
        JPanel checkboxPanel = new JPanel();
        checkboxPanel.setLayout(new BoxLayout(checkboxPanel, BoxLayout.Y_AXIS));
        checkboxPanel.setBackground(Color.WHITE);
        checkboxPanel.setBorder(BorderFactory.createEmptyBorder(30, 70, 30, 70));

        cbEducation = new JCheckBox("Education");
        cbHealth = new JCheckBox("Healthcare");
        cbDonation = new JCheckBox("Donations");

        for (JCheckBox cb : new JCheckBox[]{cbEducation, cbHealth, cbDonation}) {
            cb.setFont(new Font("Consolas", Font.PLAIN, 18));
            cb.setFocusable(false);
            cb.setBackground(Color.WHITE);
            cb.setAlignmentX(Component.LEFT_ALIGNMENT);
            checkboxPanel.add(cb);
            checkboxPanel.add(Box.createRigidArea(new Dimension(0, 10)));
        }

        add(checkboxPanel, BorderLayout.CENTER);

        // ----- BUTTON PANEL -----
        JPanel buttonPanel = new JPanel();
        buttonPanel.setBackground(Color.WHITE);

        submitButton = new JButton("Submit");
        submitButton.setFont(new Font("Segoe UI", Font.BOLD, 16));
        submitButton.setBackground(new Color(72, 201, 111));
        submitButton.setForeground(Color.WHITE);
        submitButton.setFocusPainted(false); // removes blue ring when
clicked
        submitButton.setBorderPainted(false); // removes visible border
        submitButton.setContentAreaFilled(true); // keeps the background
fill
        submitButton.setOpaque(true); // ensures solid background
        submitButton.setPreferredSize(new Dimension(120, 40));
        submitButton.addActionListener(this);

        buttonPanel.add(submitButton);
        add(buttonPanel, BorderLayout.SOUTH);

```

```

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == submitButton) {
            StringBuilder deductions = new StringBuilder("You selected: ");

            boolean hasSelection = false;
            if (cbEducation.isSelected()) {
                deductions.append("Education ");
                hasSelection = true;
            }
            if (cbHealth.isSelected()) {
                deductions.append("Healthcare ");
                hasSelection = true;
            }
            if (cbDonation.isSelected()) {
                deductions.append("Donations ");
                hasSelection = true;
            }

            if (hasSelection) {
                JOptionPane.showMessageDialog(this, deductions.toString(),
"Deductions", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Please select at least one
deduction option.", "Warning", JOptionPane.WARNING_MESSAGE);
            }

            String deductionsText = deductions.toString().trim();
            Database.set("deductions", deductionsText);
            System.out.println("Job: " + Database.getString("job"));
            System.out.println("Income: " + Database.getDouble("income"));
            System.out.println("Deductions: " + Database.getString("deductions"));

            String job = Database.getString("job");
            double income = Database.getDouble("income");
            String prompt = "My job is " + job + ". I earn " + income + " baht. My
deductions are: " + deductionsText + ". Please just calculat my total tax";

            try {
                String geminiReply = TaxCalculator.askGemini(prompt); // <--
Update method to return response
                new ResultFrame(geminiReply);
                this.dispose();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
}

```


ResultFrame.java

```
package ui;

import Services.Database;
import java.awt.*;
import javax.swing.*;

public class ResultFrame extends JFrame {

    public ResultFrame(String geminiReply) {
        setTitle("TaxHelper - Result");
        setSize(500, 600);
        setIconImage(new ImageIcon("ui/images/logo.png").getImage());
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(new BorderLayout());

        JLabel header = new JLabel("Result", SwingConstants.CENTER);
        header.setFont(new Font("Segoe UI", Font.BOLD, 24));
        header.setBorder(BorderFactory.createEmptyBorder(20, 0, 10, 0));
        add(header, BorderLayout.NORTH);

        JTextArea resultArea = new JTextArea();
        resultArea.setEditable(false);
        resultArea.setFont(new Font("Consolas", Font.PLAIN, 14));
        resultArea.setLineWrap(true);
        resultArea.setWrapStyleWord(true);

        StringBuilder content = new StringBuilder();
        content.append("Job: ").append(Database.getString("job")).append("\n");
        content.append("Income: ").append(Database.getDouble("income")).append("Baht\n");
        content.append("Deductions: ").append(Database.getString("deductions")).append("\n\n");
        content.append("----- Tax Calculation & Advice ----- \n");
        content.append(geminiReply).append("\n");

        resultArea.setText(content.toString());

        JScrollPane scrollPane = new JScrollPane(resultArea);
        scrollPane.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));
        add(scrollPane, BorderLayout.CENTER);

        setVisible(true);

        // --- Bottom Panel with Done Button ---
    }
}
```

```

        JPanel bottomPanel = new JPanel();
        bottomPanel.setBackground(Color.WHITE);
        bottomPanel.setBorder(BorderFactory.createEmptyBorder(10, 0, 20, 0));

        JButton doneButton = new JButton("Done");
        doneButton.setPreferredSize(new Dimension(120, 40));
        doneButton.setFont(new Font("Segoe UI", Font.BOLD, 16));
        doneButton.setBackground(new Color(72, 201, 111));
        doneButton.setForeground(Color.WHITE);
        doneButton.setFocusPainted(false);           // removes blue ring when
clicked
        doneButton.setBorderPainted(false);           // removes visible border
        doneButton.setContentAreaFilled(true);        // keeps the background fill
        doneButton.setOpaque(true);                   // ensures solid background
        doneButton.addActionListener(e -> animateClose());

        bottomPanel.add(doneButton);
        add(bottomPanel, BorderLayout.SOUTH);

    }

    private void animateClose() {
        Timer timer = new Timer(15, null);
        final int[] height = {getHeight()};
        final int minHeight = 0;

        timer.addActionListener(e -> {
            height[0] -= 15;

            if (height[0] <= minHeight) {
                timer.stop();
                dispose();           // close the window
                System.exit(0);      // exit the app
            } else {
                setSize(getWidth(), height[0]);
                setLocationRelativeTo(null);
            }
        });

        timer.start();
    }
}

```

Services/

Database.java

```
package Services;
```

```
import java.util.HashMap;
import java.util.Map;

public class Database {

    // This map stores key-value pairs to act as a shared memory space across
    frames
    private static final Map<String, Object> memory = new HashMap<>();

    // Store a value using a unique key
    public static void set(String key, Object value) {
        memory.put(key, value);
    }

    // Retrieve a stored value by key (returns null if not found)
    public static Object get(String key) {
        return memory.get(key);
    }

    // Retrieve a stored value as a String
    public static String getString(String key) {
        Object value = memory.get(key);

        if (value != null) {
            return value.toString();
        } else {
            return "";
        }
    }

    // Retrieve a stored value as a double (returns 0 if not found or not a
    number)
    public static double getDouble(String key) {
        Object value = memory.get(key);

        if (value instanceof Number) {
            return ((Number) value).doubleValue();
        }

        if (value != null) {
            try {
                return Double.parseDouble(value.toString());
            } catch (Exception e) {
                return 0;
            }
        }

        return 0;
    }

    // Check if a certain key has a value stored
    public static boolean contains(String key) {
        return memory.containsKey(key);
    }
}
```

```
// Clear all stored data
public static void clear() {
    memory.clear();
}
}
```

TaxCalculator.java

```
package Services;

import java.io.IOException;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class TaxCalculator {

    private static final String API_KEY = "";
    private static final String ENDPOINT =
        "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-
        flash:generateContent?key=" + API_KEY;

    public static String askGemini(String prompt) throws IOException,
        InterruptedException {
        if (API_KEY == null || API_KEY.isEmpty()) {
            return "Error: GEMINI_API_KEY not set.";
        }

        String fullPrompt = ""
        {
            "contents": [
                {
                    "role": "user",
                    "parts": [
                        {
                            "text": "You are a helpful Thai financial advisor. Give clean,
step-by-step tax calculation."
                        }
                    ]
                },
                {
                    "role": "user",
                    "parts": [
                        {
                            "text": "%s"
                        }
                    ]
                }
            ]
        }
```

```
        ]
    }
]
}
"".formatted(prompt);

HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create(ENDPOINT))
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(fullPrompt))
    .build();

HttpClient client = HttpClient.newHttpClient();
HttpResponse<String> response = client.send(request,
HttpResponse.BodyHandlers.ofString());
String body = response.body();

// Extract `"text": "` using start index and scan until unescaped closing
quote
String marker = "\"text\\\": \";
int start = body.indexOf(marker);
if (start == -1) return "Error: No text field found in Gemini response.";

start += marker.length();
StringBuilder result = new StringBuilder();
boolean escape = false;

for (int i = start; i < body.length(); i++) {
    char c = body.charAt(i);

    if (escape) {
        if (c == 'n') result.append('\n');
        else if (c == 't') result.append('\t');
        else result.append(c);
        escape = false;
    } else if (c == '\\') {
        escape = true;
    } else if (c == '"') {
        // closing quote found
        break;
    } else {
        result.append(c);
    }
}

return result.toString().trim();
}
}
```

✓ App Flow

IntroAnimation → MainFrame → IncomeFrame → DeductionFrame → ResultFrame