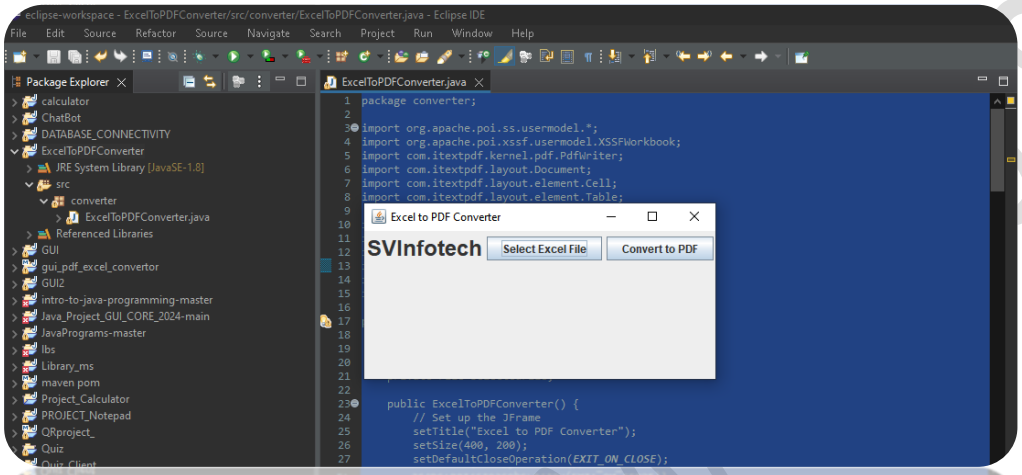
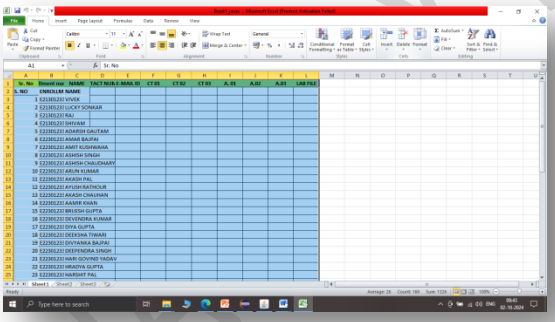
Excel to pdf file converter using Java

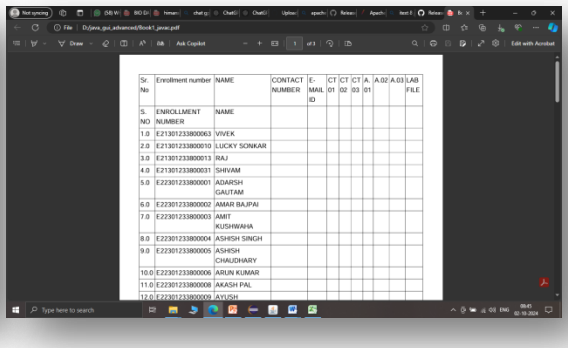
Project name- ExcelToPDFConverter

Src>> package_name-converter

Package_name >>class_name- ExcelToPDFConverter







Sr. No	Enrollment number	NAME	CONTACT NUMBER	E-MAIL ID	CT 01	CT 02	CT 03	A 01	A 02	A 03	LAB FILE
S. NO	ENROLLMENT NUMBER	NAME									
1.0	E2130123380063	VIVEK									
2.0	E2130123380010	LUCKY SONKAR									
3.0	E2130123380013	RAJ									
4.0	E2130123380031	SHIVAM									
5.0	E2230123380001	ADARSH GAUTAM									
6.0	E2230123380002	AMAR BAJPAI									
7.0	E2230123380003	AMIT KUSHWAHA									
8.0	E2230123380004	ASHISH SINGH									
9.0	E2230123380005	ASHISH CHAUDHARY									
10.0	E2230123380006	ARUN KUMAR									
11.0	E2230123380008	AKASH PAL									
12.0	E2230123380009	AYUSH									

(Excel file) ⟶ (pdf file)

Source code:

```
package converter;


import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import com.itextpdf.kernel.pdf.PdfWriter;
import com.itextpdf.layout.Document;
import com.itextpdf.layout.element.Cell;
import com.itextpdf.layout.element.Table;


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


public class ExcelToPDFConverter extends JFrame
 {
```

```
    private JButton selectExcelButton;
    private JButton convertButton;
    private File selectedFile;

    public ExcelToPDFConverter()
{
        // Set up the JFrame
        setTitle("Excel to PDF Converter by svinfotech himanshu singh");
        setSize(400, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

        // Create and add title label
        JLabel titleLabel = new JLabel("SVInfotech");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
 // Set font and size
        titleLabel.setHorizontalAlignment(SwingConstants.CENTER);
// Center the text
        add(titleLabel);
```

```
        // Create and add buttons
        selectExcelButton = new JButton("Select Excel File");
        convertButton = new JButton("Convert to PDF");

        add(selectExcelButton);
        add(convertButton);

        // Button to select an Excel file
        selectExcelButton.addActionListener(new ActionListener()
{
            @Override
            public void actionPerformed(ActionEvent e)
{
                JFileChooser fileChooser = new JFileChooser();
                int result = fileChooser.showOpenDialog(null);
                if (result == JFileChooser.APPROVE_OPTION) {
                    selectedFile = fileChooser.getSelectedFile();
                    JOptionPane.showMessageDialog(null, "Selected File: " +
selectedFile.getName());
                }
```

```
        }
    });

    // Button to convert the selected Excel file to PDF
    convertButton.addActionListener(new ActionListener()
{
        @Override
        public void actionPerformed(ActionEvent e)
 {
            if (selectedFile != null) {
                try {
                    convertExcelToPDF(selectedFile);
                    JOptionPane.showMessageDialog(null, "Conversion
Successful!");
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(null, "Error during
conversion: " + ex.getMessage());
                }
            } else {
                JOptionPane.showMessageDialog(null, "Please select an
Excel file first.");
```

```
            }

        }

    });

  }


  public void convertExcelToPDF(File excelFile) throws Exception

{

    // Load the Excel workbook

    FileInputStream fis = new FileInputStream(excelFile);

    Workbook workbook = new XSSFWorkbook(fis);

    Sheet sheet = workbook.getSheetAt(0);


    // Create a new PDF document

    String pdfFileName = excelFile.getAbsolutePath().replace(".xlsx",
".pdf");

    PdfWriter writer = new PdfWriter(pdfFileName);

    com.itextpdf.kernel.pdf.PdfDocument pdfDoc = new
com.itextpdf.kernel.pdf.PdfDocument(writer);

    Document document = new Document(pdfDoc);
```

```
// Create a table with the number of columns equal to the number of columns in the first row of the Excel sheet
int numberOfColumns = sheet.getRow(0).getPhysicalNumberOfCells();
Table table = new Table(numberOfColumns);


// Iterate over rows and cells to add data to the PDF table
for (int rowIndex = 0; rowIndex <= sheet.getLastRowNum(); rowIndex++) {
    Row row = sheet.getRow(rowIndex);
    if (row != null) {
        for (int colIndex = 0; colIndex < row.getPhysicalNumberOfCells(); colIndex++) {
            Cell cell = new Cell().add(new com.itextpdf.layout.element.Paragraph(row.getCell(colIndex).toString()));
            table.addCell(cell);
        }
    }
}


// Add the table to the PDF document
```

```
        document.add(table);


        // Close the workbook and the document
        workbook.close();
        document.close();
    }

    public static void main(String[] args) {
        // Run the GUI application
        SwingUtilities.invokeLater(() -> {
            ExcelToPDFConverter converter = new ExcelToPDFConverter();
            converter.setVisible(true);
        });
    }
}
```

Create this project in eclipse ide:

Run this project with alt+shift+x

Add jar files of apache poi version 5 and itext version 8

Link to download jar files:

Poi jar files-

https://dlcdn.apache.org/poi/release/src/apache-poi-src-5.3.0-20240625.zip

above are for excel files.

itext jar files-

https://github.com/itext/itext-java/releases/download/8.0.5/iText-Core-8.0.5-only-jars.zip

above are for pdf files.

How to add external jars in project:

Right click on project from project explorer>>build path>>configure build path>>select libraries tab>>add external jar>>select jars>> click apply >> click apply and close