

## Code Documentation

### Packages Used

#### **os**

Provides functions for interacting with the operating system. Used to create directories if they don't exist.

#### **pandas (pd)**

A powerful data manipulation and analysis library for Python. Used to read and process the Excel file containing employee data.

#### **fpdf (FPDF)**

A library to create PDF documents in Python. Used to create the PDF format for the payslip.

#### **tkinter (tk)**

A standard GUI (Graphical User Interface) toolkit in Python. Used to create the graphical user interface for selecting the Excel file and entering the employee ID.

#### **tkinter.filedialog**

Provides dialogs for file selection. Used to open a dialog box for selecting the Excel file.

#### **tkinter.messagebox**

Provides a way to show message boxes. Used to show error and success messages.

## Code Documentation

### Functions and Classes

#### Class PDF(FPDF)

Inherits from the FPDF class to customize the PDF format for the payslip.

#### Methods

header(self): Adds a header to the PDF, including a logo and company information. Called automatically when a new page is added to the PDF.

footer(self): Adds a footer to the PDF with the page number. Called automatically at the end of each page in the PDF.

#### Function create\_payslip(employee)

Creates a PDF payslip for a given employee.

#### Parameters

employee (Series): A pandas Series object containing the employee's data.

#### Steps

- Create a PDF object.
- Add a page and set up the title and various sections.

## Code Documentation

- Fill in the employee details and the classified income and deductions tables.
- Add totals and footer information.

### Returns

A PDF object with the formatted payslip.

### Function `generate_payslip()`

Handles the process of generating the payslip from selecting the Excel file to creating the PDF.

### Steps

- Open a file dialog to select the Excel file.
- Read the Excel file into a DataFrame.
- Check if all required columns are present.
- Get the employee ID from the user input.
- Locate the employee data in the DataFrame.
- Call the `create_payslip` function to generate the PDF.

## Code Documentation

- Save the PDF to the specified directory.
- Show success or error messages as appropriate.

### GUI Setup

- `root = tk.Tk()`: Initializes the main window of the GUI.
- `root.title("Payslip Generator")`: Sets the title of the main window.
- `tk.Label(root, text="Enter Employee ID:").pack(pady=5)`: Adds a label to the GUI.
- `employee_id_entry = tk.Entry(root)`: Adds an entry widget for the user to input the employee ID.
- `employee_id_entry.pack(pady=5)`: Packs the entry widget into the main window.
- `tk.Button(root, text="Generate Payslip", command=generate_payslip).pack(pady=20)`: Adds a button to the GUI that triggers the `generate_payslip` function when clicked.
- `root.mainloop()`: Starts the GUI event loop.