

Payslip Generator Documentation

Introduction

A payslip, also known as a pay stub, is a document issued by an employer to an employee that outlines the employee's earnings, deductions, and net pay for a specific period. This payslip generator application allows users to generate a detailed payslip in PDF format based on data from an Excel file. The application uses Python libraries such as `gradio`, `pandas`, and `fpdf` to achieve this functionality.

Table of Contents

1. Introduction
2. Required Libraries
3. Code Breakdown
 - Class Definition: PDF
 - Function: `create_payslip`
 - Function: `generate_payslip`
 - Function: `download_payslip`
 - Gradio Interface
4. How to Use the Application
5. Conclusion

Required Libraries

The application requires the following Python libraries:

1. `gradio`: For creating the user interface.
2. `pandas`: For reading and manipulating Excel data.
3. `fpdf`: For generating PDF documents.
4. `os`: For file path operations.

python

Copy code

```
import gradio as gr
import pandas as pd
from fpdf import FPDF
```

```
import os
```

Code Breakdown

Class Definition: PDF

The `PDF` class inherits from `FPDF` and customizes the header and footer of the PDF document.

python

Copy code

```
class PDF(FPDF):
    def header(self):
        self.set_font("Arial", 'B', 12)
        self.cell(0, 5, "SYMBIOSYS TECHNOLOGIES", ln=True, align='C')
        self.cell(0, 5, "Plot No 1&2, Hill no-2, IT Park,", ln=True,
align='C')
        self.cell(0, 5, "Rushikonda, Visakhapatnam-45", ln=True,
align='C')
        self.cell(0, 5, "Ph: 2550369, 2595657", ln=True, align='C')
        self.ln(10)

    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', 'I', 8)
        self.cell(0, 10, f'Page {self.page_no()}', 0, 0, 'C')
```

Function: `create_payslip`

The `create_payslip` function generates the payslip PDF for a given employee.

python

Copy code

```
def create_payslip(employee):
    pdf = PDF()
    pdf.add_page()

    # Add the payslip title
    pdf.set_font("Arial", 'B', 12)
```

```
pdf.cell(0, 10, "SALARY STATEMENT FOR THE MONTH OF "
f"{employee['Statement for the month']}" " 2024", ln=True, align='C')
pdf.ln(5)
```

```
# Table 1: Employee Code, Name, Designation
```

```
pdf.set_font("Arial", 'B', 10)
pdf.cell(65, 8, "Employee Code", border=1)
pdf.cell(65, 8, "Employee Name", border=1)
pdf.cell(65, 8, "Designation", border=1)
pdf.ln()
pdf.set_font("Arial", size=10)
pdf.cell(65, 8, f"{employee['Employee Code']}", border=1)
pdf.cell(65, 8, f"{employee['Employee Name']}", border=1)
pdf.cell(65, 8, f"{employee['Designation']}", border=1)
pdf.ln(10)
```

```
# Table 2: Date of Joining, Employment Status, Statement for the
month
```

```
pdf.set_font("Arial", 'B', 10)
pdf.cell(65, 8, "Date of Joining", border=1)
pdf.cell(65, 8, "Employment Status", border=1)
pdf.cell(65, 8, "Statement for the month", border=1)
pdf.ln()
pdf.set_font("Arial", size=10)
date_of_joining = pd.to_datetime(employee['Date of
Joining']).strftime('%d-%m-%Y')
pdf.cell(65, 8, f"{date_of_joining}", border=1)
pdf.cell(65, 8, f"{employee['Employment Status']}", border=1)
pdf.cell(65, 8, f"{employee['Statement for the month']}",
border=1)
pdf.ln(10)
```

```
# Table 3 and 4: Classified Income and Deductions side by side
```

```
pdf.set_font("Arial", 'B', 10)
pdf.cell(70, 8, "Classified Income", border=1, align='C')
pdf.cell(30, 8, "Amount (Rs.)", border=1, align='C')
pdf.cell(60, 8, "Deductions", border=1, align='C')
pdf.cell(30, 8, "Amount (Rs.)", border=1, align='C')
```

```

pdf.ln()
pdf.set_font("Arial", size=10)

income_items = [
    "Basic Pay (Rs.)", "House Rent Allowance (Rs.)",
    "City Compensatory Allowance (Rs.)", "Travel Allowance (Rs.)",
    "Food Allowance (Rs.)", "Performance Incentives (Rs.)"
]
deduction_items = [
    "Professional Tax (Rs.)", "Income Tax (Rs.)",
    "Provident Fund (Rs.)", "ESI (Rs.)",
    "Leaves-Loss of Pay (Rs.)", "Others (Rs.)"
]

for income, deduction in zip(income_items, deduction_items):
    pdf.cell(70, 8, f"{income.replace('(Rs.)', '').strip()}",
border=1)
    pdf.cell(30, 8, f"Rs. {employee[income]:.2f}", border=1,
align='R')
    pdf.cell(60, 8, f"{deduction.replace('(Rs.)', '').strip()}",
border=1)
    pdf.cell(30, 8, f"Rs. {employee[deduction]:.2f}", border=1,
align='R')
    pdf.ln()

# Add spacing before the Totals section
pdf.ln(10)

# Totals section
pdf.set_font("Arial", 'B', 10)
pdf.cell(70, 8, "GROSS PAY", border=1)
pdf.cell(30, 8, f"Rs. {employee['Gross Pay (Rs.)']:.2f}",
border=1, align='R')
pdf.cell(60, 8, "DEDUCTIONS", border=1)
pdf.cell(30, 8, f"Rs. {employee['Deductions (Rs.)']:.2f}",
border=1, align='R')
pdf.ln()
pdf.cell(100, 8, "NET PAY", border=1)

```

```

        pdf.cell(80, 8, f"Rs. {employee['Net Pay (Rs.)']:.2f}", border=1,
align='R')
        pdf.ln(20)

        # Footer section with added spacing
        pdf.cell(0, 8, "AUTHORIZED SIGNATORY", ln=True)
        pdf.ln(20) # Added more spacing here
        pdf.cell(0, 8, "Durgaaprasadh,", ln=True)
        pdf.cell(0, 8, "H.R Executive", ln=True)
        pdf.ln(10)
        pdf.set_font('Arial', 'I', 8)
        pdf.cell(0, 8, "We request you to verify employment details with
our office on email: hr@symbiosystech.com. (+91-0891-2550369)",
ln=True)

    return pdf

```

Function: generate_payslip

The `generate_payslip` function reads employee data from an Excel file and generates the payslip PDF for a specific employee.

python

Copy code

```

def generate_payslip(file, emp_id):
    df = pd.read_excel(file.name)
    emp_id = int(emp_id)
    employee = df.loc[df['Employee Code'] == emp_id].squeeze()

    if not employee.empty:
        payslip_pdf = create_payslip(employee)
        output_filename = f"payslip_{emp_id}.pdf"
        payslip_pdf.output(output_filename)
        return output_filename
    else:
        return "Employee ID not found in the uploaded file."

```

Function: `download_payslip`

The `download_payslip` function facilitates the downloading of the generated payslip PDF.

python

Copy code

```
def download_payslip(file, emp_id):
    payslip_file = generate_payslip(file, emp_id)
    if payslip_file != "Employee ID not found in the uploaded file.":
        return payslip_file
    else:
        return None
```

Gradio Interface

The Gradio interface defines the user interface for the payslip generator application.

python

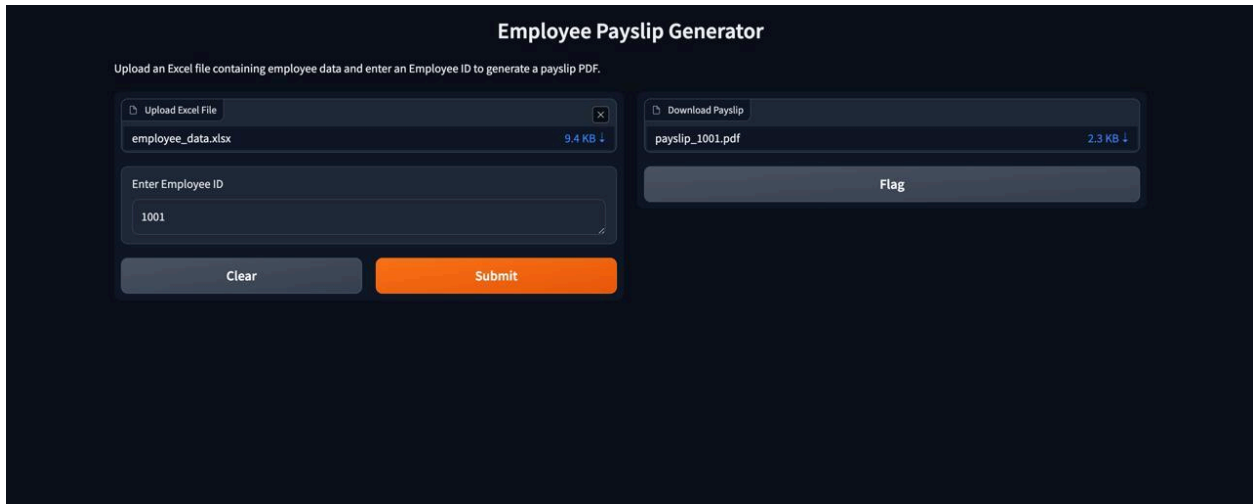
Copy code

```
iface = gr.Interface(
    fn=download_payslip,
    inputs=[gr.File(label="Upload Excel File"),
            gr.Textbox(label="Enter Employee ID")],
    outputs=gr.File(label="Download Payslip"),
    title="Employee Payslip Generator",
    description="Upload an Excel file containing employee data and enter an Employee ID to generate a payslip PDF."
)

# Launch the Gradio interface with sharing enabled
iface.launch(share=True)
```

How to Use the Application

1. Upload an Excel file containing employee data.
2. Enter the Employee ID for which you want to generate the payslip.
3. Click on the "Download Payslip" button to download the generated payslip PDF.



The screenshot displays the 'Employee Payslip Generator' web application. At the top, a title bar reads 'Employee Payslip Generator'. Below it, a subtitle states: 'Upload an Excel file containing employee data and enter an Employee ID to generate a payslip PDF.' The interface is divided into two main sections. The left section contains an 'Upload Excel File' button, a file input field showing 'employee_data.xlsx' with a '9.4 KB' size indicator, an 'Enter Employee ID' text box with '1001' entered, and two buttons: 'Clear' and 'Submit'. The right section features a 'Download Payslip' button, a file output field showing 'payslip_1001.pdf' with a '2.3 KB' size indicator, and a 'Flag' button.

Conclusion

This payslip generator application provides a simple and efficient way to generate payslips in PDF format. By leveraging the capabilities of `gradio`, `pandas`, and `fpdf`, it ensures accurate and professional documentation of employee earnings and deductions.

Output

SYMBIOSYS TECHNOLOGIES
Plot No 1&2, Hill no-2, IT Park,
Rushikonda, Visakhapatnam-45
Ph: 2550369, 2595657

SALARY STATEMENT FOR THE MONTH OF JULY 2024

| | | |
|-----------------|-------------------|-------------------------|
| Employee Code | Employee Name | Designation |
| 1001 | Sriram | Software Engineer |
| Date of Joining | Employment Status | Statement for the month |
| 01-01-2023 | Full Time | JULY |

| | | | |
|-----------------------------|--------------|--------------------|--------------|
| Classified Income | Amount (Rs.) | Deductions | Amount (Rs.) |
| Basic Pay | Rs. 50000.00 | Professional Tax | Rs. 0.00 |
| House Rent Allowance | Rs. 15000.00 | Income Tax | Rs. 0.00 |
| City Compensatory Allowance | Rs. 8000.00 | Provident Fund | Rs. 0.00 |
| Travel Allowance | Rs. 2000.00 | ESI | Rs. 0.00 |
| Food Allowance | Rs. 1500.00 | Leaves-Loss of Pay | Rs. 0.00 |
| Performance Incentives | Rs. 3000.00 | Others | Rs. 0.00 |

| | | | |
|-----------|--------------|------------|--------------|
| GROSS PAY | Rs. 99500.00 | DEDUCTIONS | Rs. 0.00 |
| NET PAY | | | Rs. 99500.00 |

AUTHORISED SIGNATORY

Durgaaprasadh,
H.R Executive

We request you to verify employment details with our office on email: hr@symbiosystech.com. (+91-0891-2550369)

