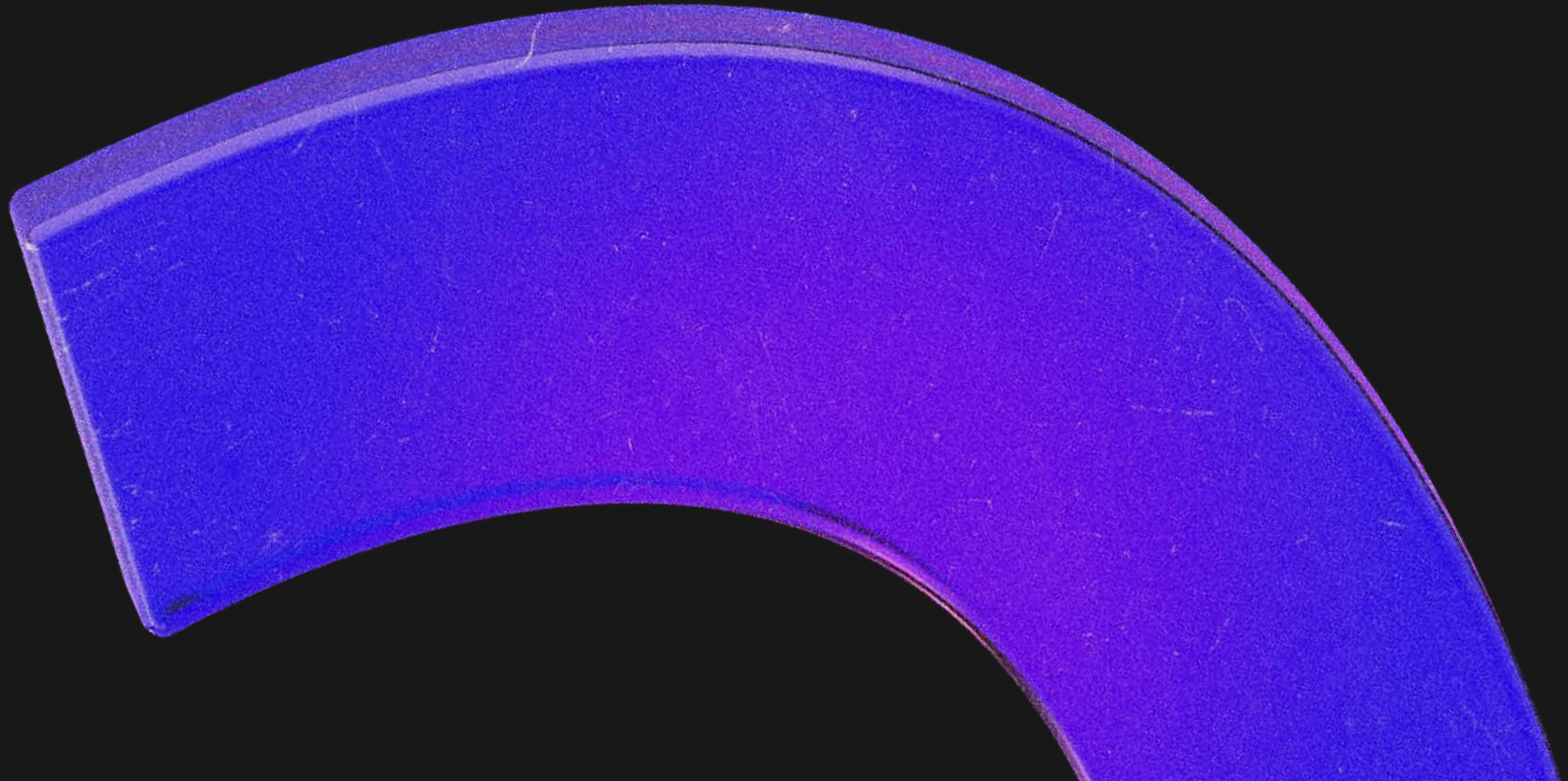
Employee Payroll Project Python(DOIT)

Student: Sergo Shaginovi

Tbilisi, 2024



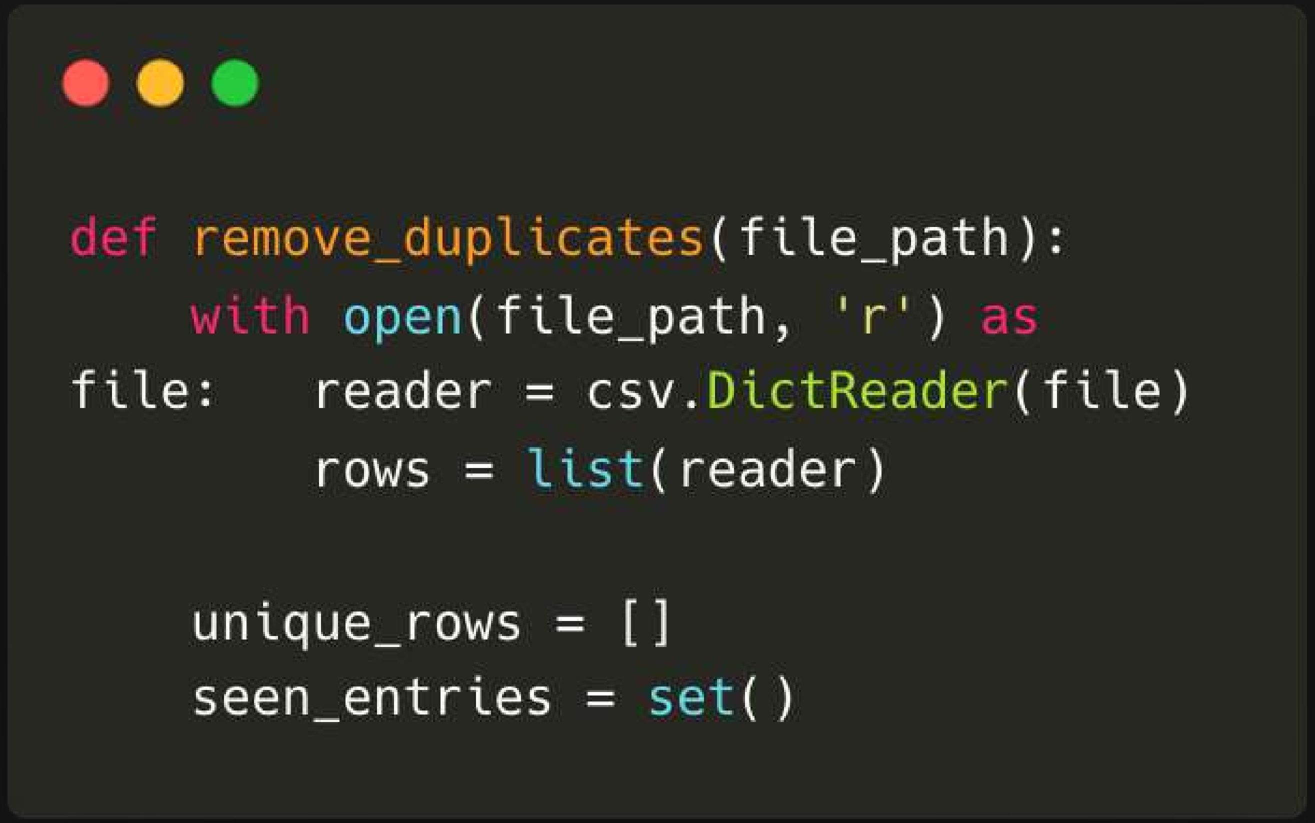
Main Steps

We are working with two files:
'employees.csv' and 'salaries.csv'(Salaries file has to contain some information)

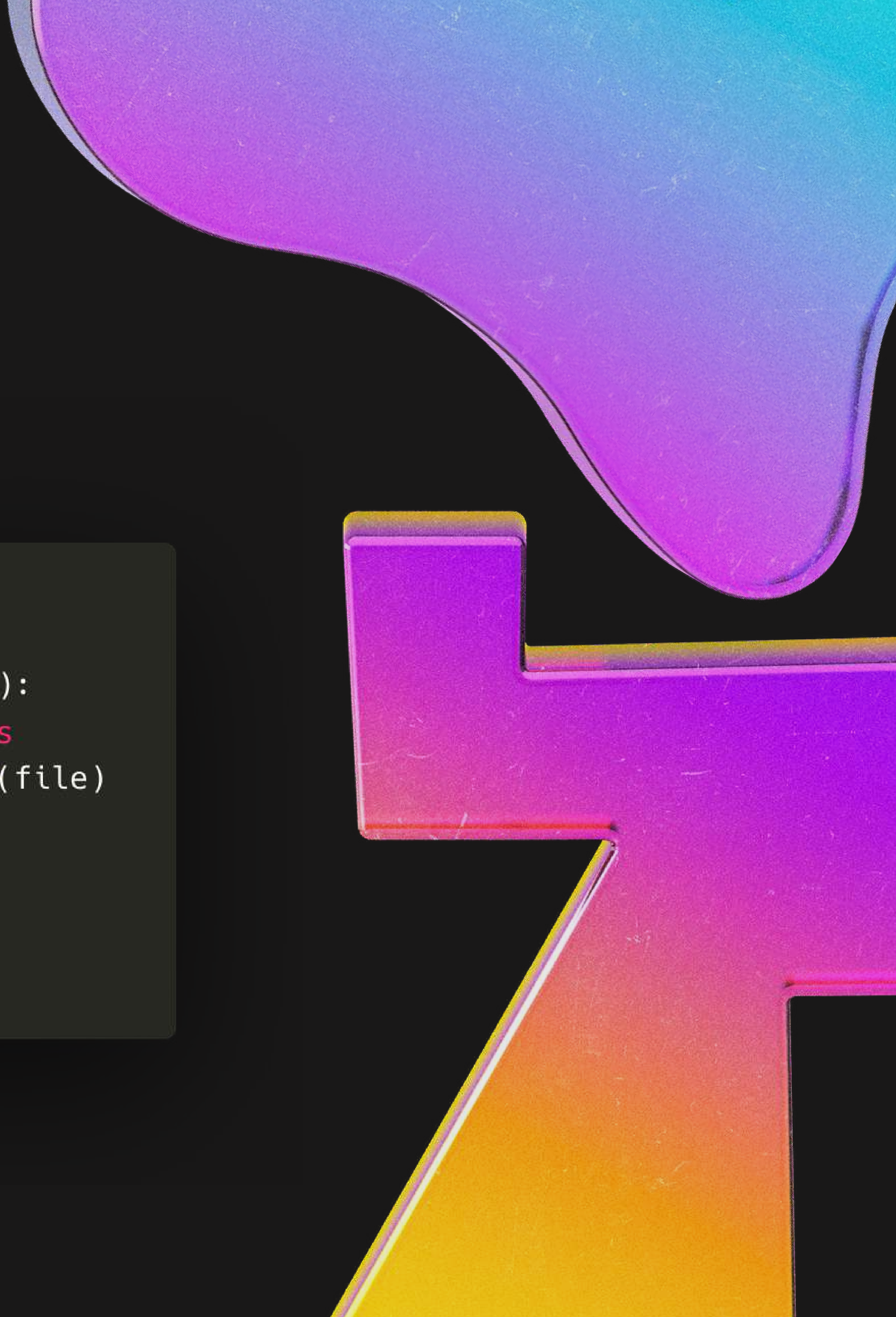
- Removing possible dublicates from employees.csv
- Asking user if there are any new employees to be added
- Add missing information
- Update Salaries Information

Remove Duplicates (remove_duplicates function)

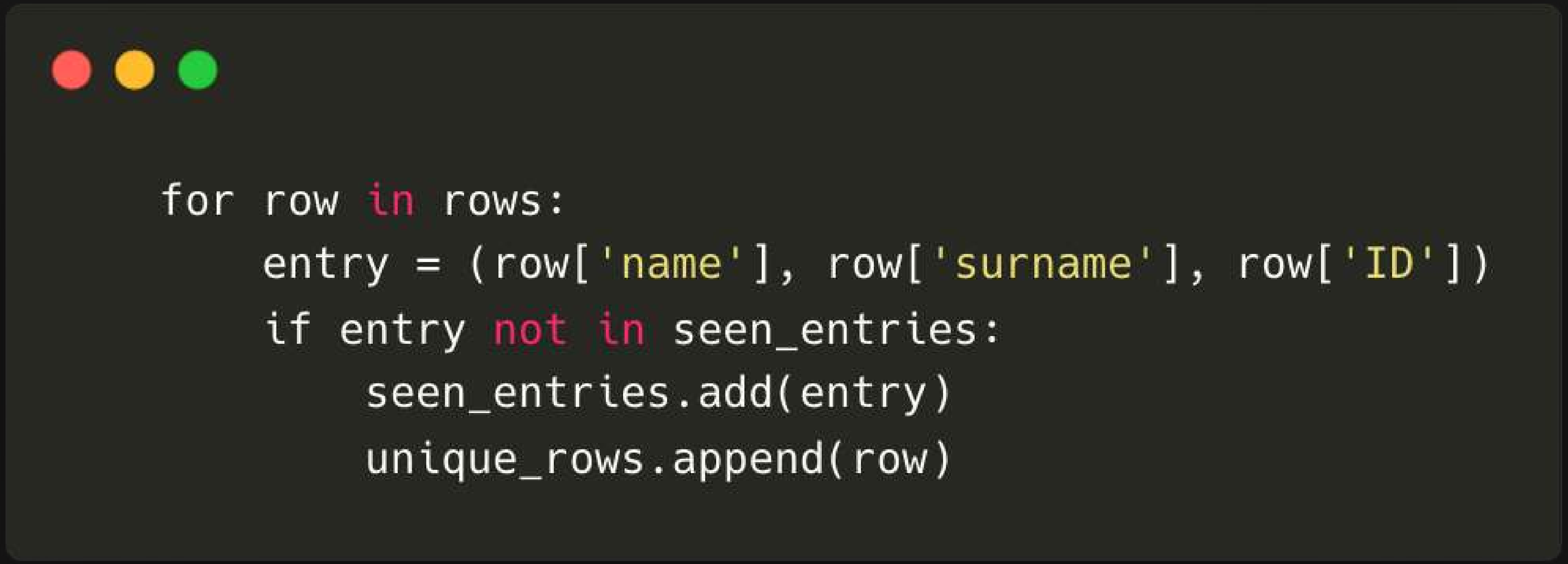
- Reads data from the employees.csv file using the csv.DictReader to create a list of dictionaries (rows).



```
def remove_duplicates(file_path):
    with open(file_path, 'r') as
file:    reader = csv.DictReader(file)
        rows = list(reader)

    unique_rows = []
    seen_entries = set()
```

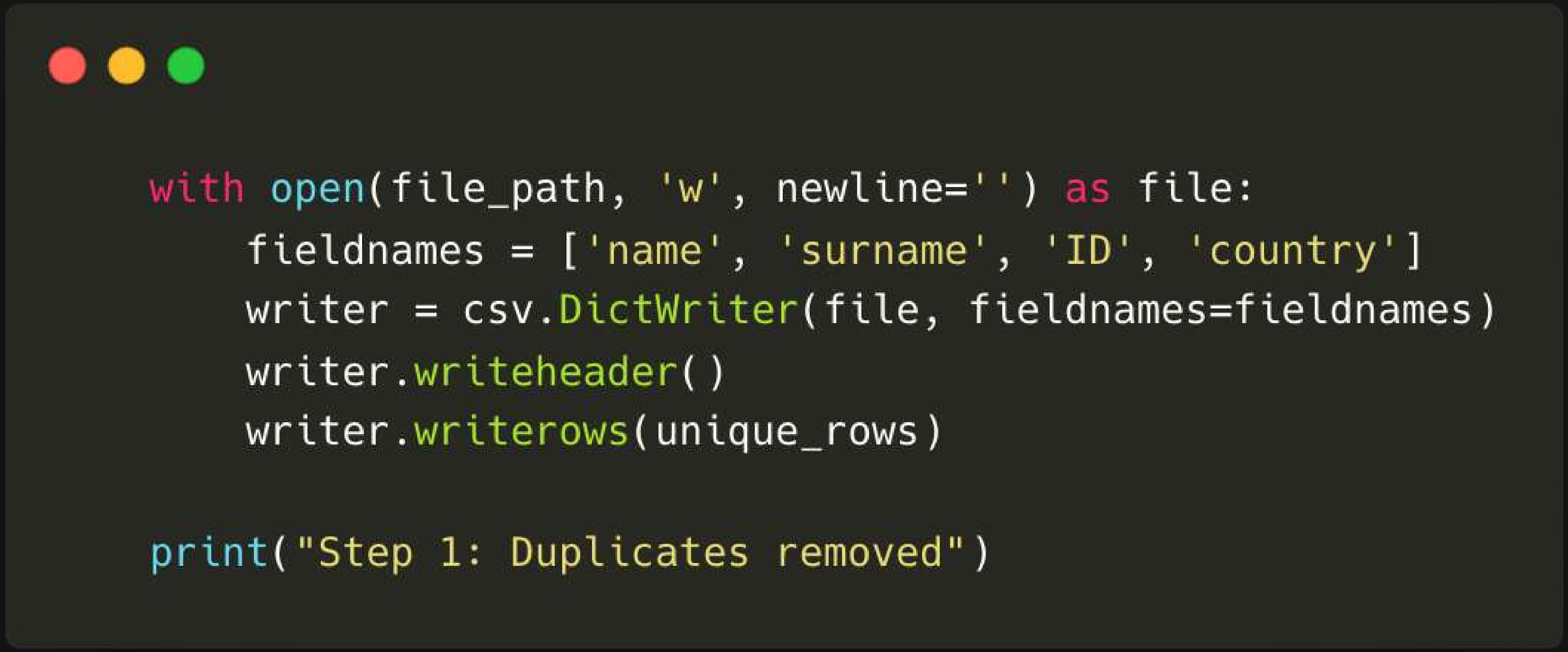


- Iterates through the list of dictionaries, keeping track of unique entries based on name, surname, and ID.



```python
for row in rows:
    entry = (row['name'], row['surname'], row['ID'])
    if entry not in seen_entries:
        seen_entries.add(entry)
        unique_rows.append(row)
```

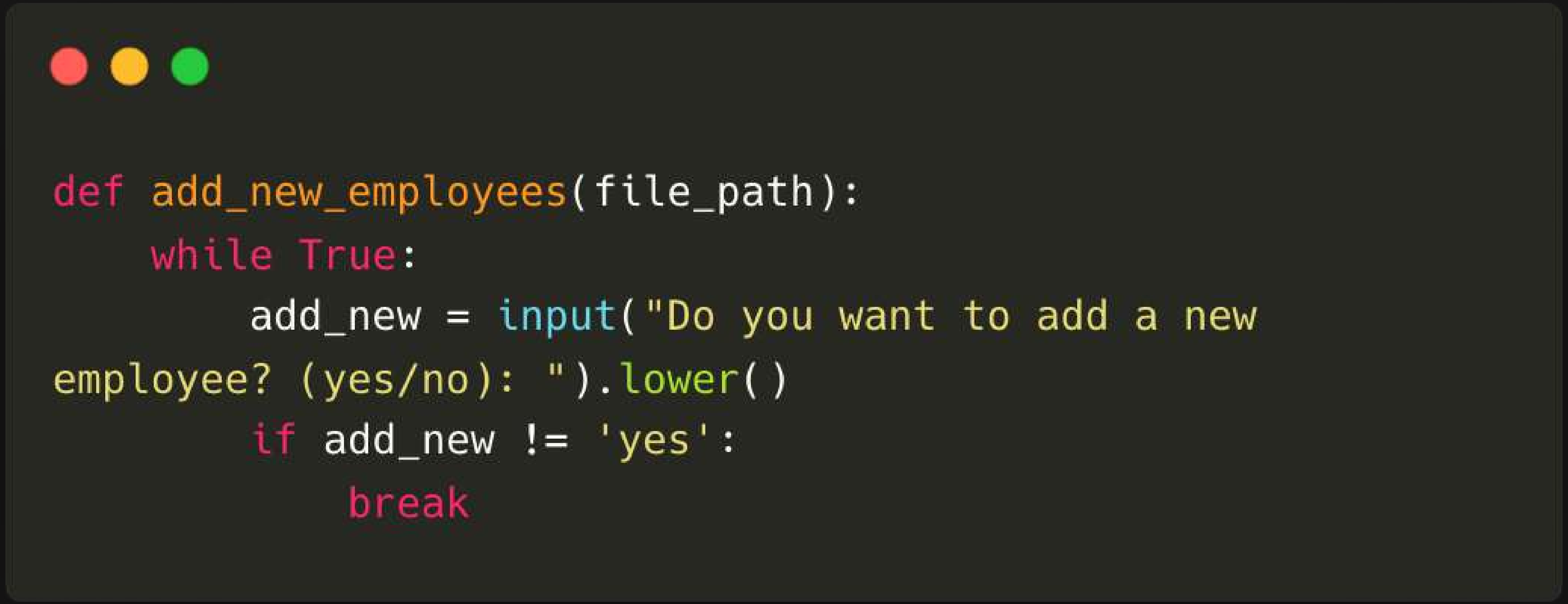
- Writes the unique entries back to the employees.csv file, effectively removing duplicates.
- Prints a message indicating that duplicates have been removed.



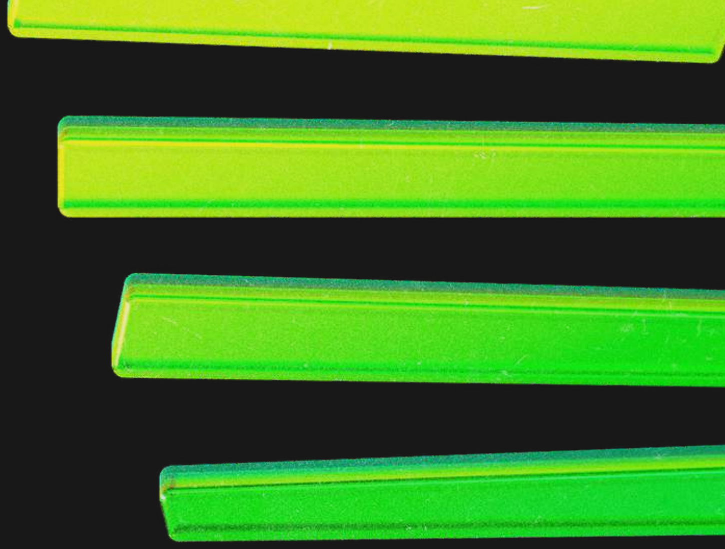
```python
with open(file_path, 'w', newline='') as file:
    fieldnames = ['name', 'surname', 'ID', 'country']
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(unique_rows)

print("Step 1: Duplicates removed")
```

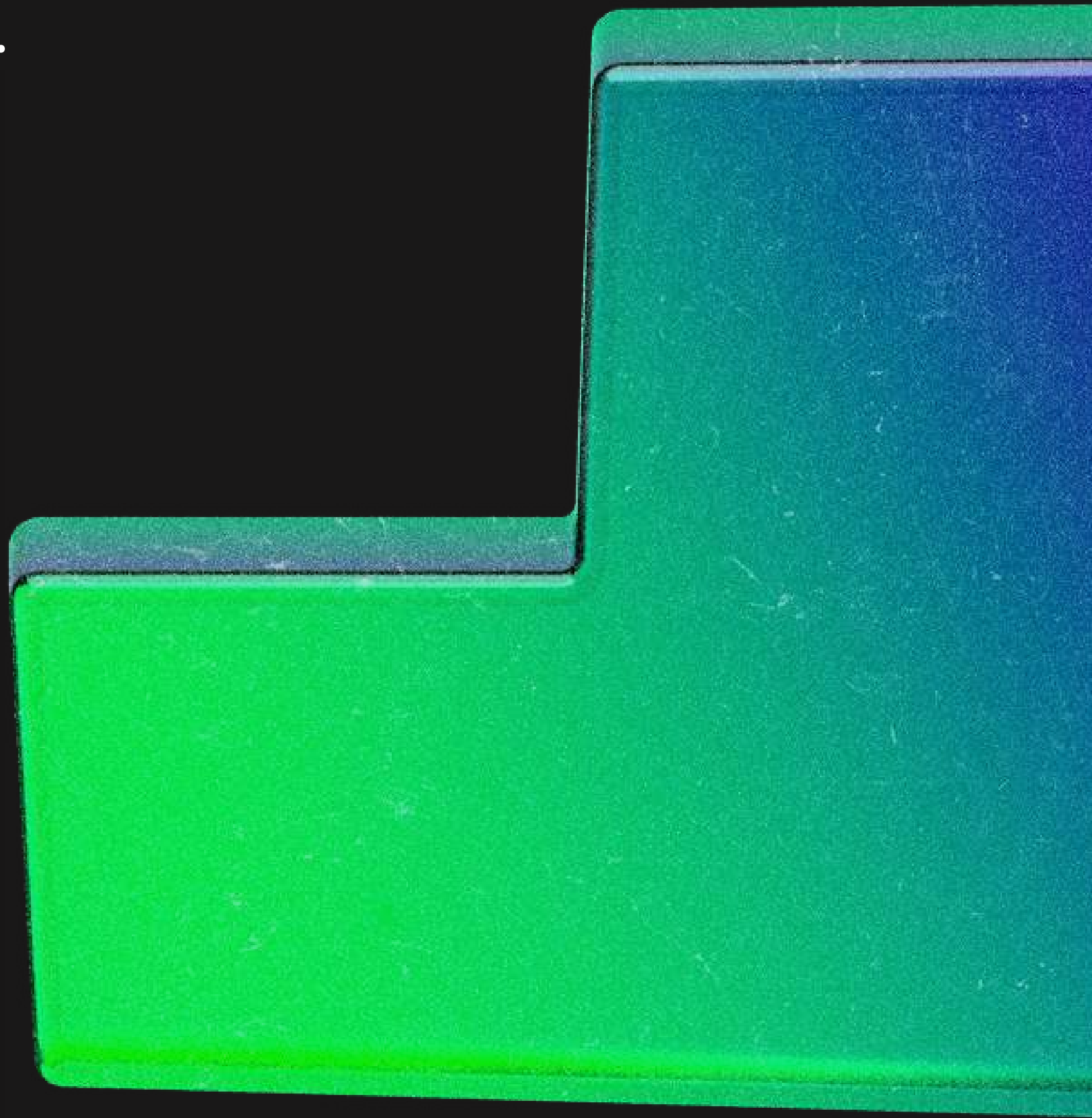
Add New Employees (add_new_employees function):

- Takes user input to determine if new employees need to be added.

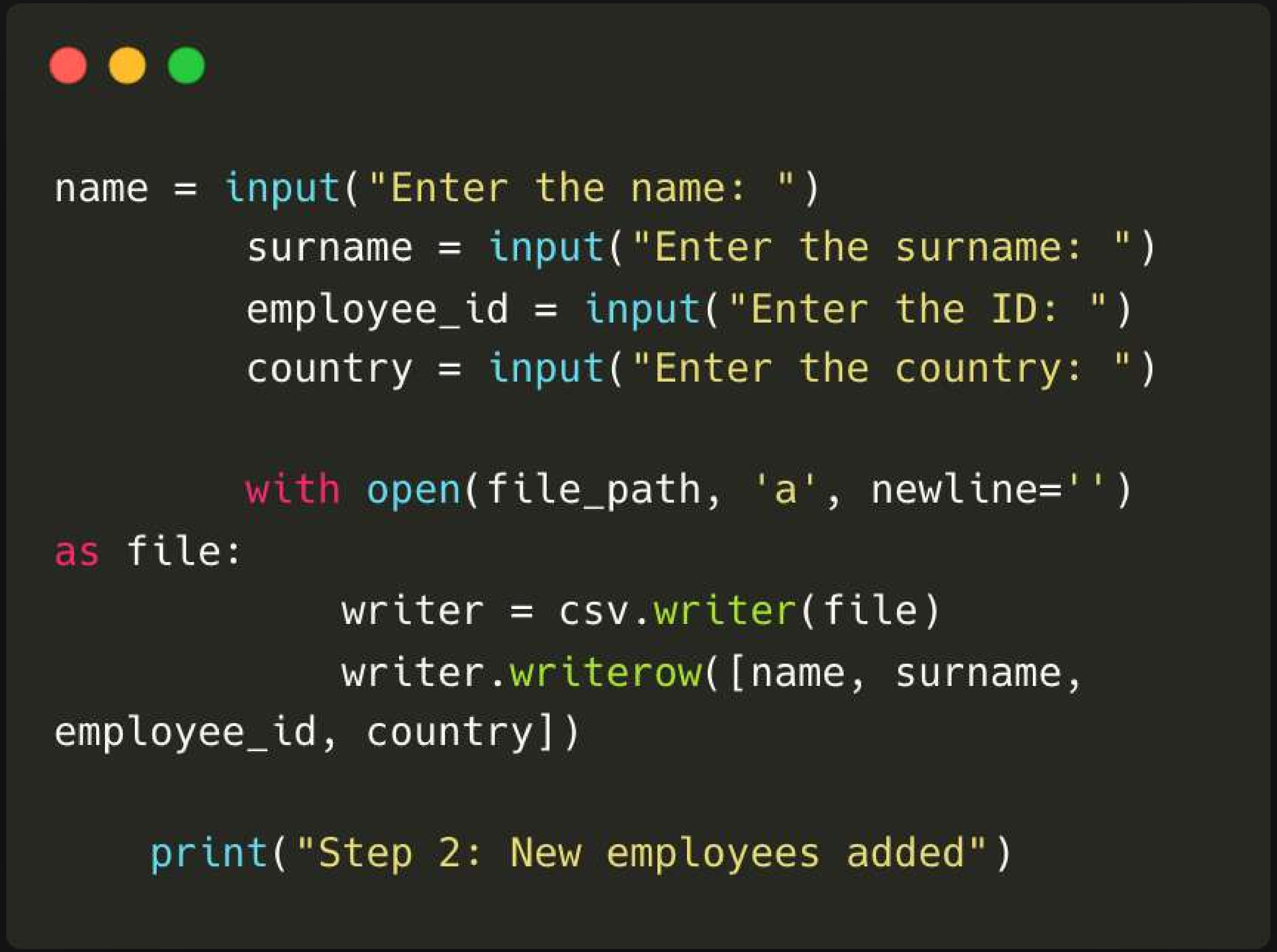


```
def add_new_employees(file_path):
    while True:
        add_new = input("Do you want to add a new
employee? (yes/no): ").lower()
        if add_new != 'yes':
            break
```





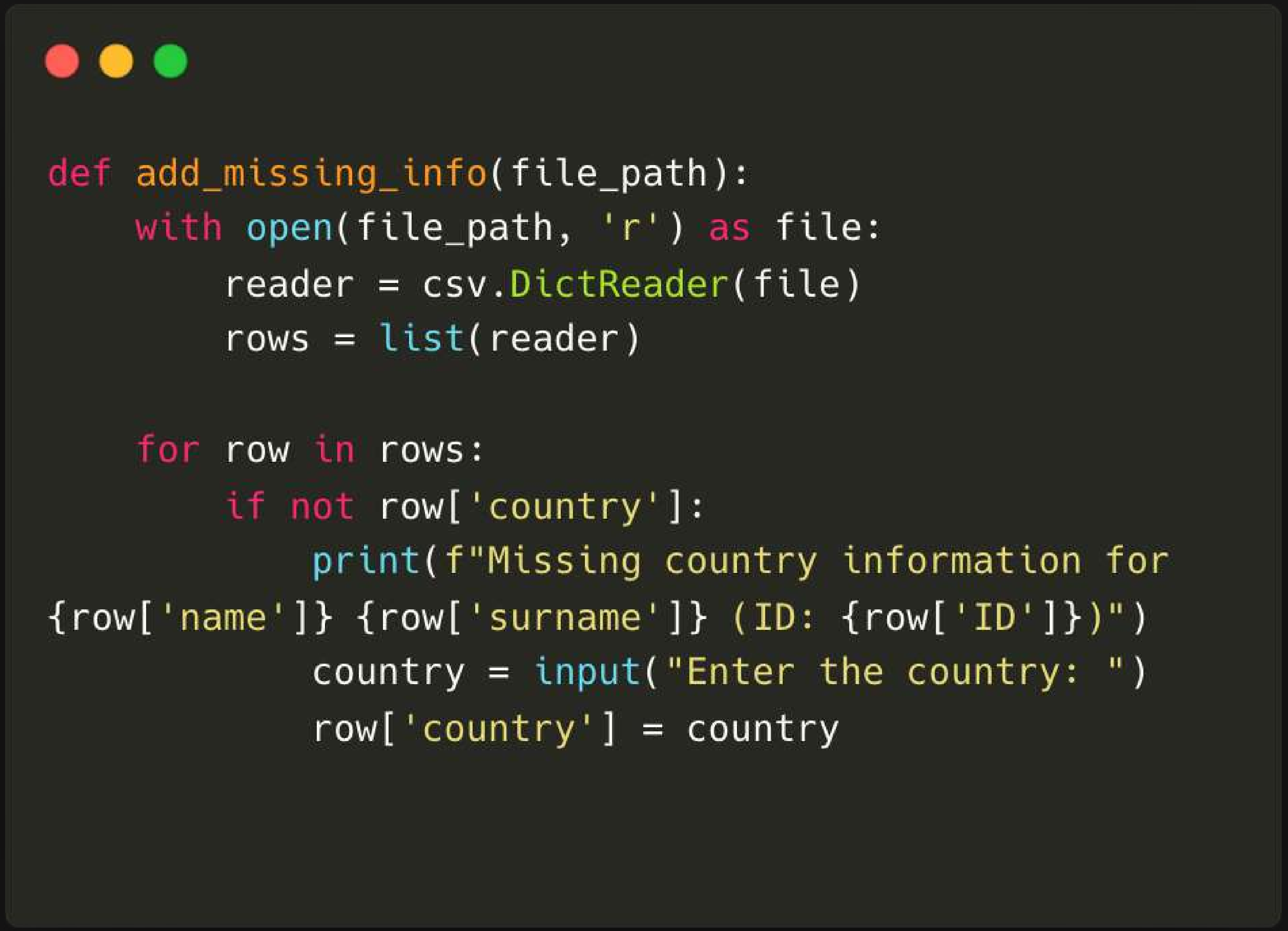
- If yes, takes user input for name, surname, ID, and country, and creates Employee objects.
- Appends these new employees to the employees.csv file.
- Prints a message indicating that new employees have been added.



```
name = input("Enter the name: ")
        surname = input("Enter the surname: ")
        employee_id = input("Enter the ID: ")
        country = input("Enter the country: ")

        with open(file_path, 'a', newline='')
as file:
            writer = csv.writer(file)
            writer.writerow([name, surname,
employee_id, country])

    print("Step 2: New employees added")
```

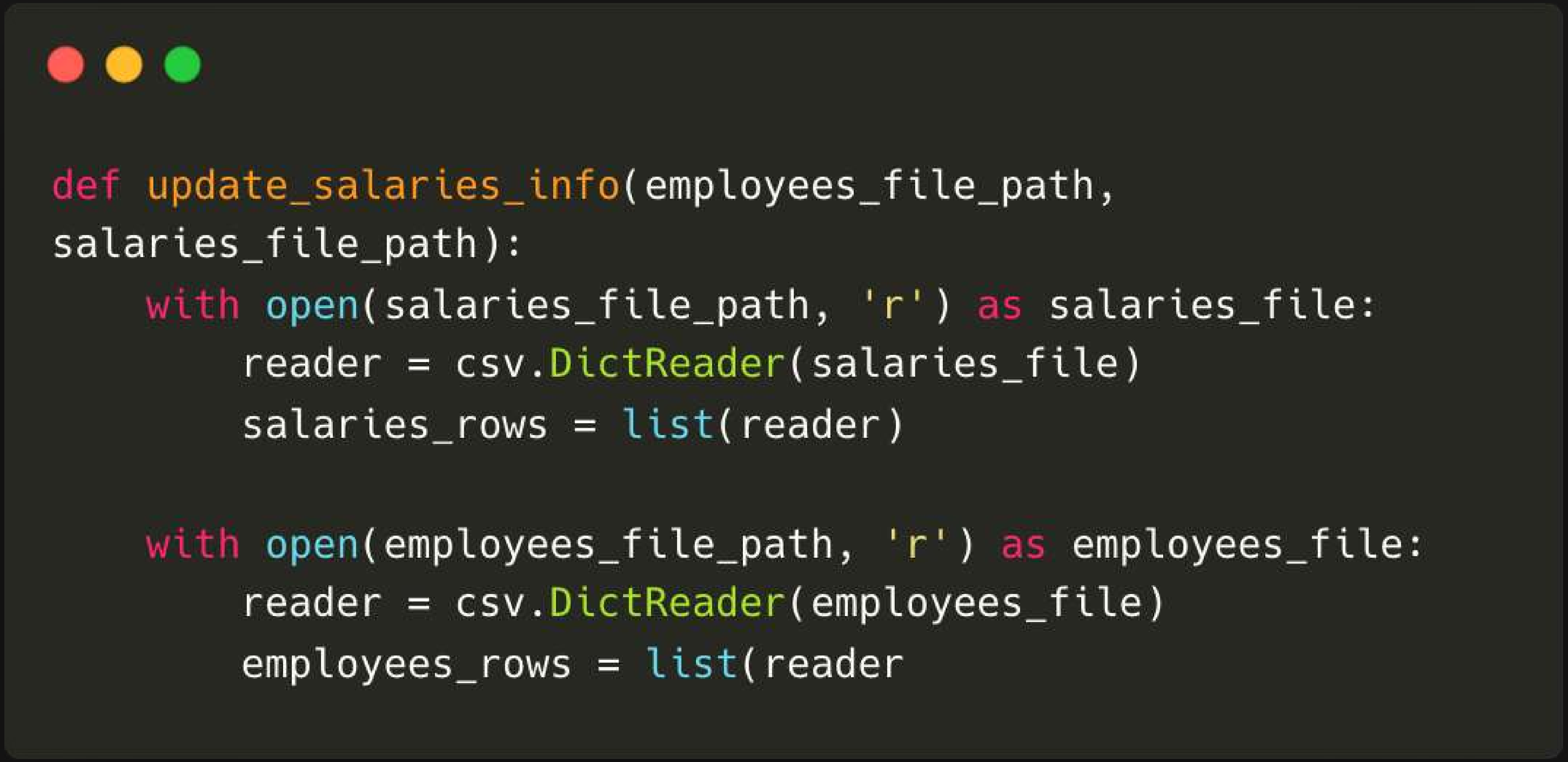
Add Missing Information (add_missing_info function):

- Reads data from the employees.csv file using csv.DictReader.
- Iterates through the employees to check for missing country information.
- If country information is missing, prompts the user for input and updates the file.



```
def add_missing_info(file_path):
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        rows = list(reader)

    for row in rows:
        if not row['country']:
            print(f"Missing country information for
{row['name']} {row['surname']} (ID: {row['ID']})")
            country = input("Enter the country: ")
            row['country'] = country
```

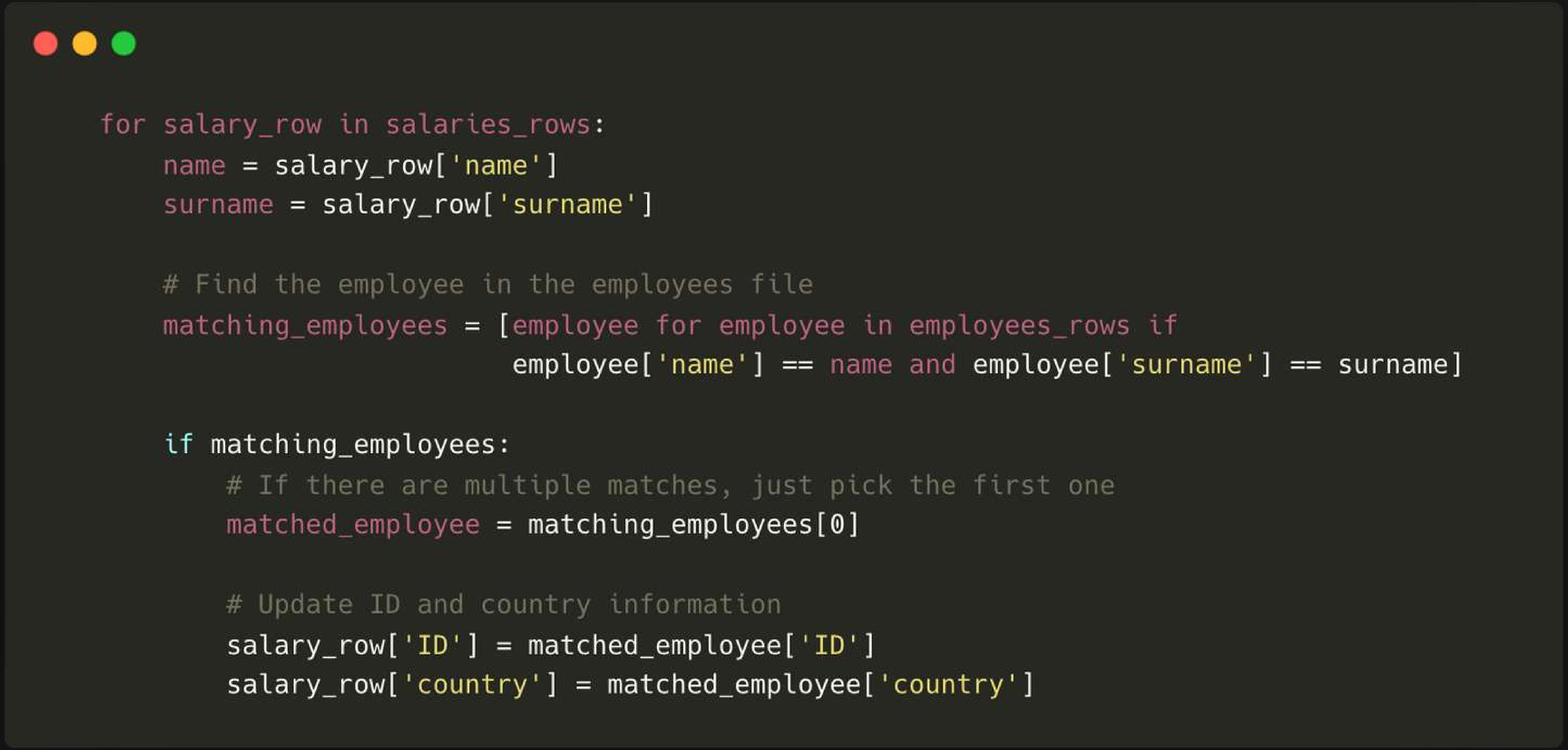
Update Salaries Information (update_salaries_info function):



```
def update_salaries_info(employees_file_path,
salaries_file_path):
    with open(salaries_file_path, 'r') as salaries_file:
        reader = csv.DictReader(salaries_file)
        salaries_rows = list(reader)

    with open(employees_file_path, 'r') as employees_file:
        reader = csv.DictReader(employees_file)
        employees_rows = list(reader
```

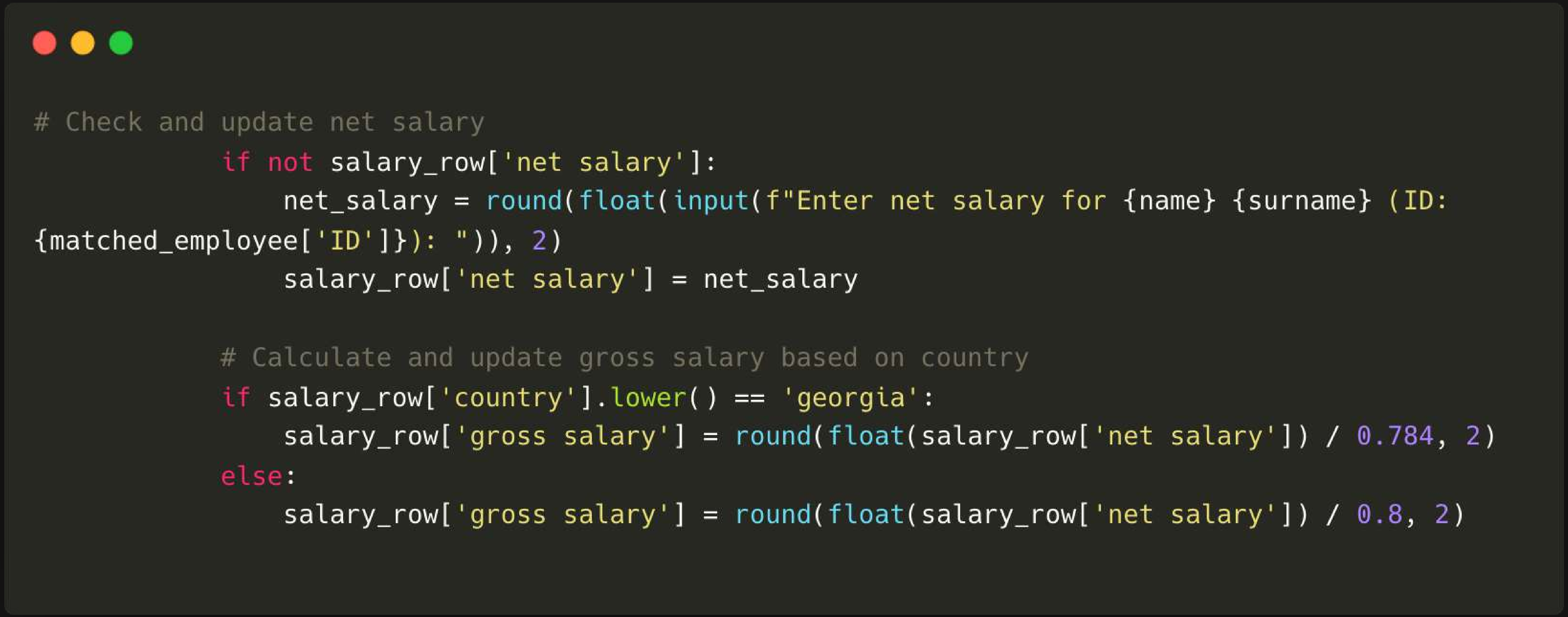
- Reads data from the salaries.csv file and employees.csv file.

- Iterates through the salaries data and matches it with employees based on name and surname.

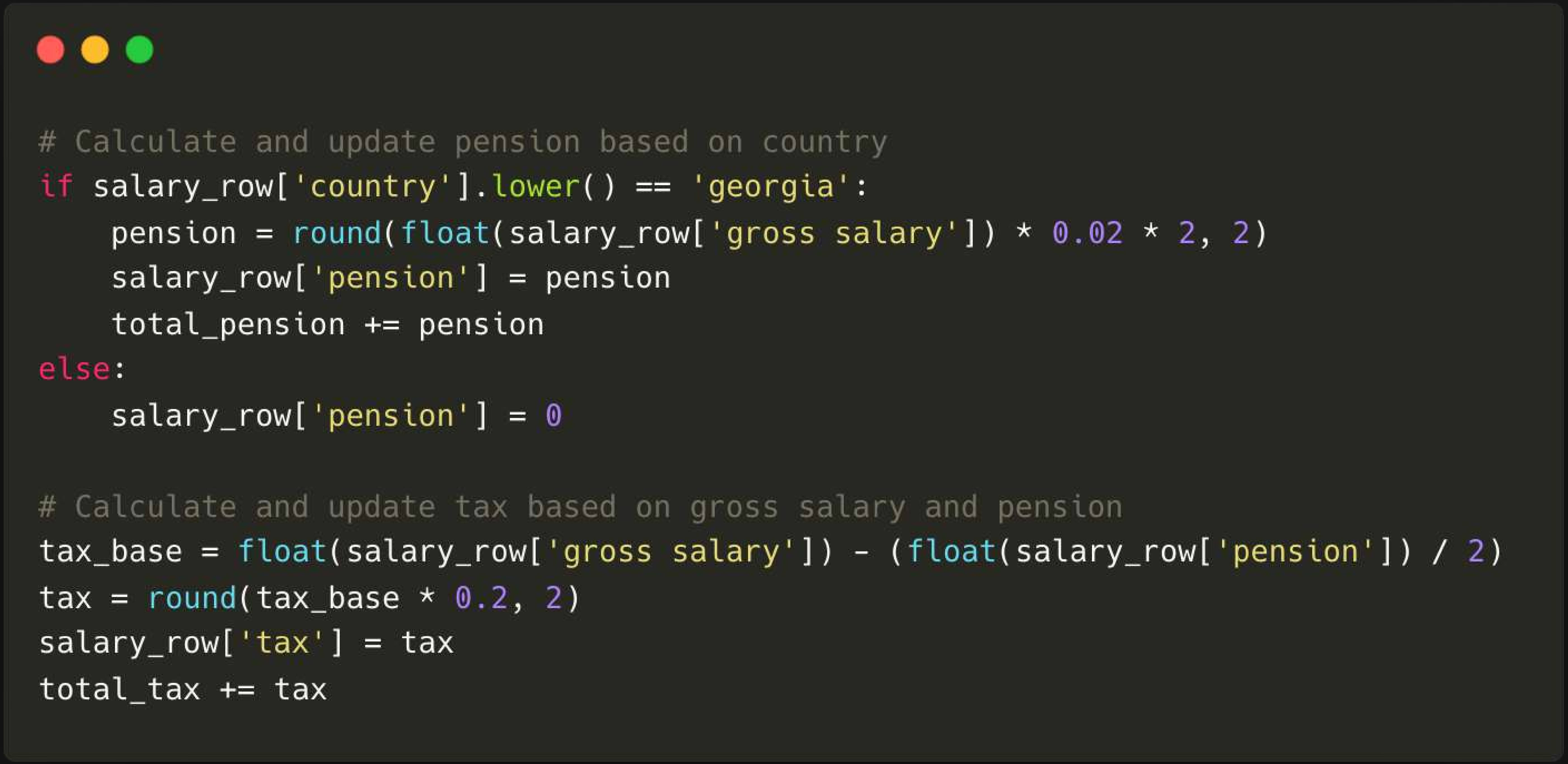


```python
for salary_row in salaries_rows:
    name = salary_row['name']
    surname = salary_row['surname']

    # Find the employee in the employees file
    matching_employees = [employee for employee in employees_rows if
                          employee['name'] == name and employee['surname'] == surname]

    if matching_employees:
        # If there are multiple matches, just pick the first one
        matched_employee = matching_employees[0]

        # Update ID and country information
        salary_row['ID'] = matched_employee['ID']
        salary_row['country'] = matched_employee['country']
```

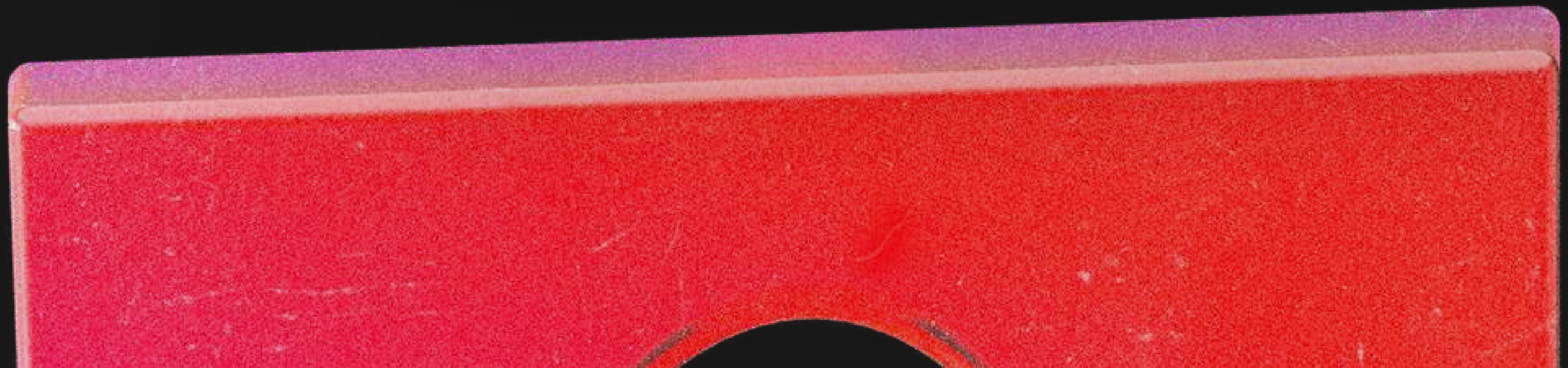
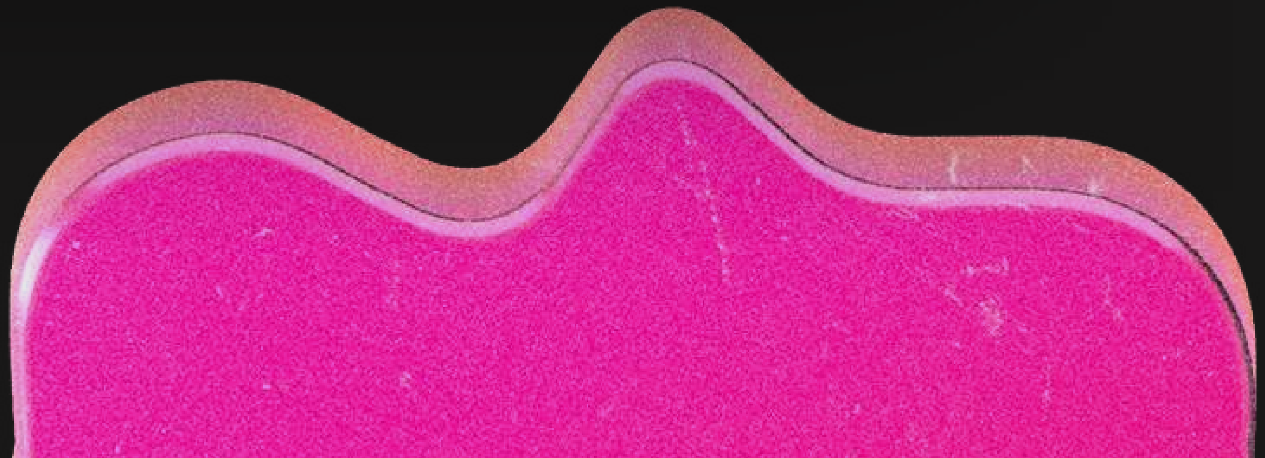
- Calculates and updates the gross salary, pension, and tax based on provided rules. (Residents of Georgia - 20% PIT tax + 4% Pensions, others - 20% PIT tax, no pension tax)



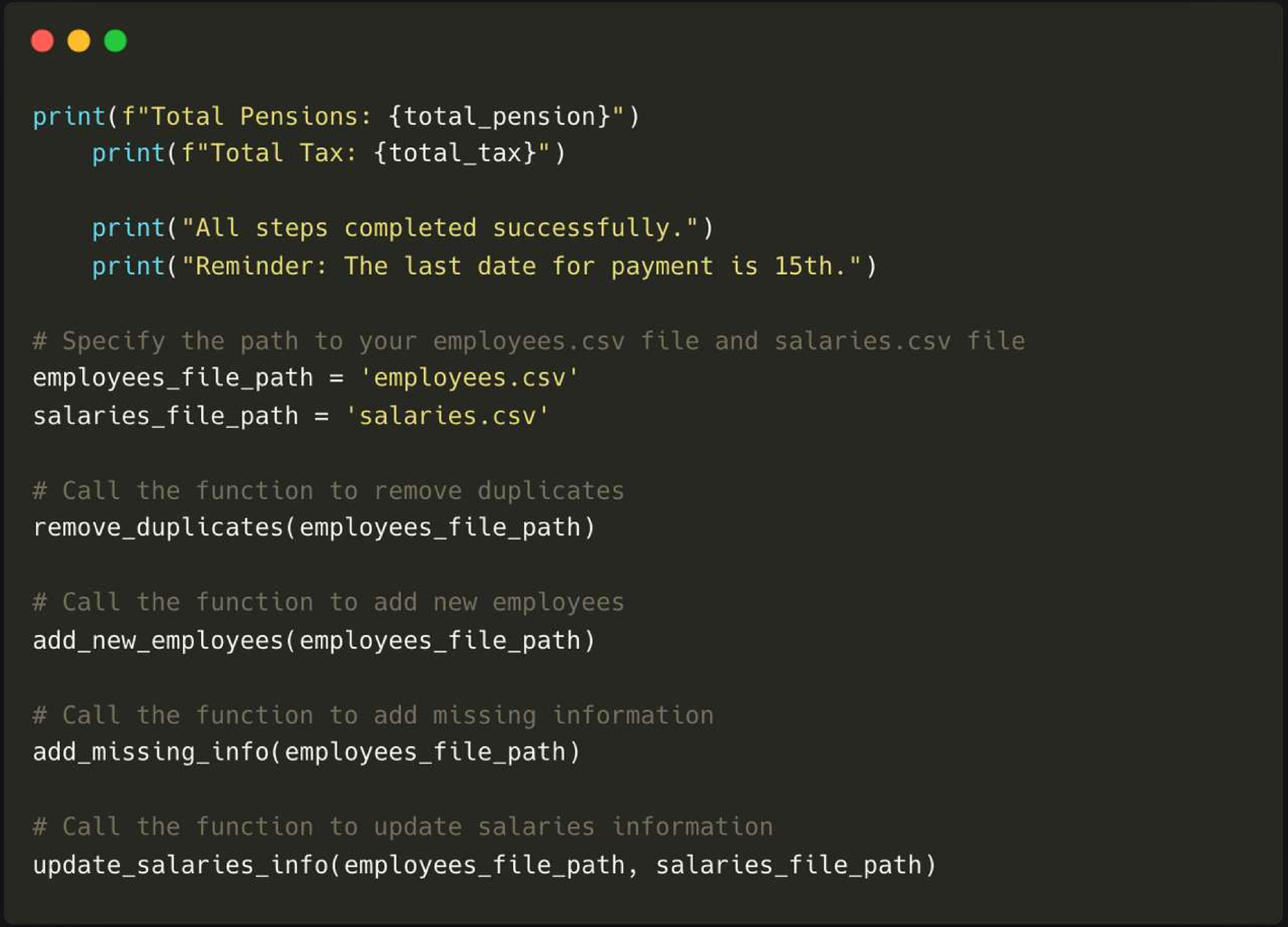
```
# Check and update net salary
      if not salary_row['net salary']:
        net_salary = round(float(input(f"Enter net salary for {name} {surname} (ID:
{matched_employee['ID']}): ")), 2)
        salary_row['net salary'] = net_salary

      # Calculate and update gross salary based on country
      if salary_row['country'].lower() == 'georgia':
        salary_row['gross salary'] = round(float(salary_row['net salary']) / 0.784, 2)
      else:
        salary_row['gross salary'] = round(float(salary_row['net salary']) / 0.8, 2)
```



```python
# Calculate and update pension based on country
if salary_row['country'].lower() == 'georgia':
    pension = round(float(salary_row['gross salary']) * 0.02 * 2, 2)
    salary_row['pension'] = pension
    total_pension += pension
else:
    salary_row['pension'] = 0

# Calculate and update tax based on gross salary and pension
tax_base = float(salary_row['gross salary']) - (float(salary_row['pension']) / 2)
tax = round(tax_base * 0.2, 2)
salary_row['tax'] = tax
total_tax += tax
```

Final Step



```python
print(f"Total Pensions: {total_pension}")
    print(f"Total Tax: {total_tax}")

    print("All steps completed successfully.")
    print("Reminder: The last date for payment is 15th.")

# Specify the path to your employees.csv file and salaries.csv file
employees_file_path = 'employees.csv'
salaries_file_path = 'salaries.csv'

# Call the function to remove duplicates
remove_duplicates(employees_file_path)

# Call the function to add new employees
add_new_employees(employees_file_path)

# Call the function to add missing information
add_missing_info(employees_file_path)

# Call the function to update salaries information
update_salaries_info(employees_file_path, salaries_file_path)
```

Thanks for attention!

I’m open to any questions!