



UNIVERSITI TEKNOLOGI MARA

KEDAH BRANCH

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS

IML208: PROGRAMMING FOR LIBRARIES

ASSESSMENT 1:

INDIVIDUAL PROJECT.

Prepared by:

NURUL ANIS ADEELA BINTI ROSELE (2022606634)

GROUP KCDIM1443F

Prepared for:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

Submission date:

Week 12

INDIVIDUAL PROJECT.

PREPARED BY:

NURUL ANIS ADEELA BINTI ROSELE (2022606634)

GROUP KCDIM1443F

CDIM144 – DIPLOMA IN LIBRARY INFORMATICS

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UiTM)

KEDAH BRANCH



STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

Cheating: Using or attempt to use any unauthorized device, assistance, sources, practice, or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.

Plagiarism: Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.

Fabrication: Falsifying data, information, or citations in any academic assessment and evaluation.

Deception: Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.

Furnishing false information: Providing false information or false representation to any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in any manner, violate this pledge.

Name: NURUL ANIS ADEELA BINTI ROSELE

Matric Number: 2022606634

Course Code: IML208

Programme Code: CDIM144

Faculty / Campus: UiTM Kampus Sungai Petani

ACKNOWLEDGEMENT

Assalamualaikum w.b.t to Sir Airul Shazwan Bin Norshahimi. First, I would like to express my gratitude to my lecturer, Sir Airul Shazwan Bin Norshahimi, for giving me the amazing chance to finish this assignment. My first assignment for IML 208 is this one. I am aiming for a flawless grade.

I expressed my gratitude to God for granting me the capacity to think and breathe, which enabled me to finish my assignment. Living a long life will be a good gift for everyone. I want to express my gratitude to my family especially my parents for raising me with love and grace so that I could pursue my studies at UiTM Kedah.

Not to mention, my classmates have been incredibly helpful to me outside of class in getting my work done. Having a wonderful classmate who can assist me in the classroom and beyond makes me happy.

TABLE OF CONTENT

| | |
|---------------------------------|------------|
| ACKNOWLEDGEMENT | i |
| 1.0 INTRODUCTION | 1 |
| 2.0 FLOWCHART | 2 |
| 3.0 SNAPSHOT OF CODE | 3-4 |
| 4.0 SNAPSHOT OF GUI | 5 |
| 5.0 SNAPSHOT OF DATABASE | 6 |
| 6.0 CONCLUSION | 7 |

1.0 INTRODUCTION

This 'Employee Management System' offers a straightforward and user-friendly interface for managing critical personnel data. It was developed with Python and Tkinter. The application aims to simplify the process of entering and managing employee details with colorful buttons and drop-down menus.

The system is designed for ease of use, ensuring accessibility for users with different computer experience levels. It prevents mistakes by verifying accurate information like Employee ID, Name, and Monthly Salary. Although it doesn't currently connect to a database, it can be connected in the future. Users can choose options from dropdown menus for Department and Job Title, and the system updates job title options based on the chosen department, providing the most relevant choices.

The system allows users to add employees by providing details such as their ID, name, department, job title, years of experience, and monthly pay. Users can choose from dropdown menus and update job title options as they choose different departments. The system then calculates important details like annual salary and total earnings by clicking the "Add Employee" button.

Since it is designed to be adaptable, this employee management system is a useful tool for maintaining organization when handling employee data.

2.0 FLOWCHART

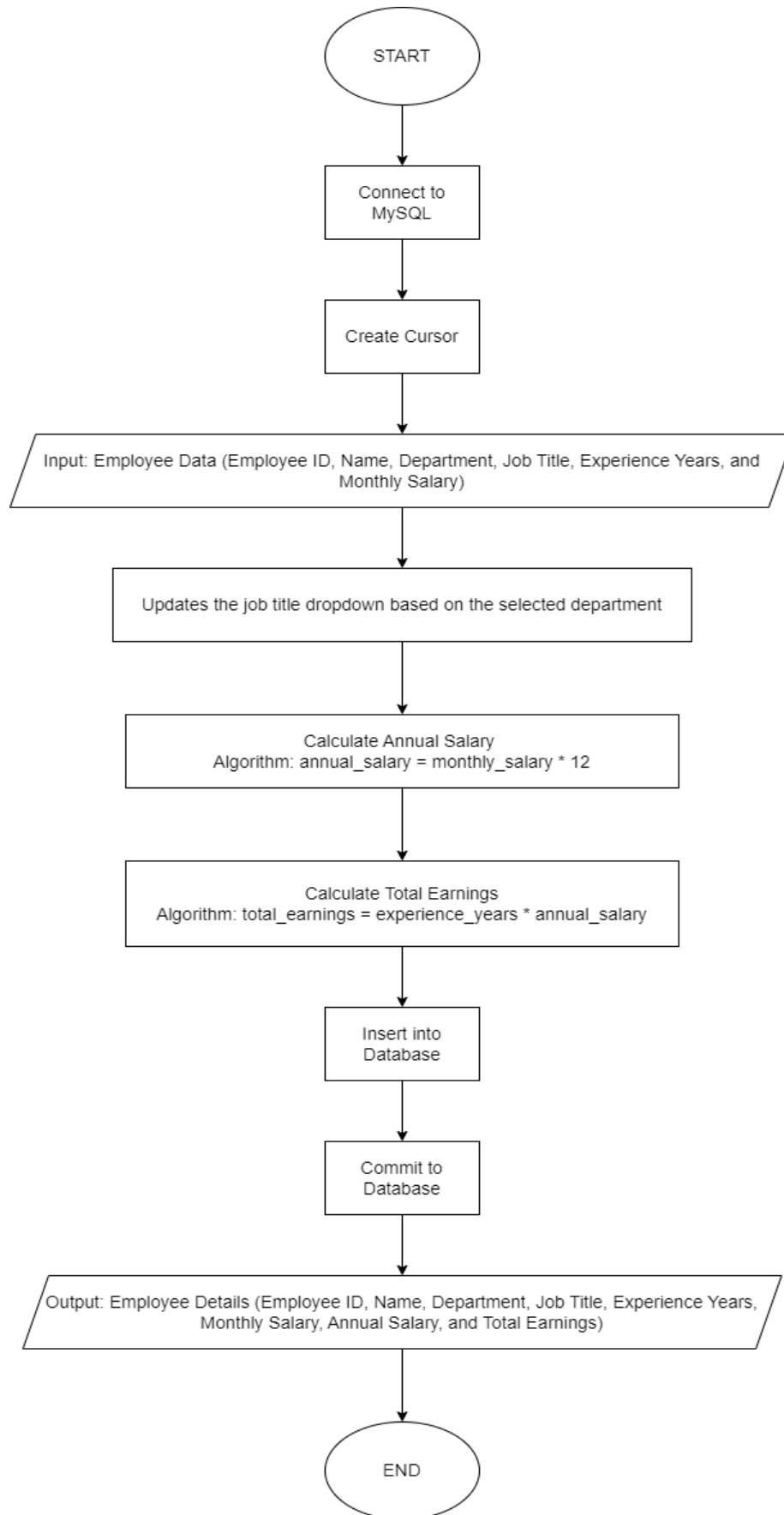


Figure 1 Flowchart Employee Management

3.0 SNAPSHOT OF CODE

SOURCE CODE > Individual Assignment 208 > employee_management.py > ...

```
1 import tkinter as tk
2 import tkinter.ttk as ttk
3 import mysql.connector
4
5 # Connect to your MySQL database
6 mydb = mysql.connector.connect(
7     host="localhost",
8     user="root",
9     password="",
10    database="employee_management"
11 )
12
13 # Create a cursor object to execute SQL queries
14 mycursor = mydb.cursor()
15
16 # Function to handle the employee addition and database saving
17 def add_employee():
18     employee_id = int(entry_id.get())
19     name = entry_name.get()
20     department = department_var.get()
21     job_title = job_title_var.get()
22     experience_years = int(entry_experience.get())
23     monthly_salary = float(entry_salary.get())
24
25     # Calculate annual salary
26     annual_salary = monthly_salary * 12
27
28     # Calculate total earnings based on experience years
29     total_earnings = experience_years * annual_salary
30
31     # To insert your Data into your database
32     sql = "INSERT INTO `employees` (Employee_ID, Employee_Name, Employee_Department, Job_Title, Experience_Years, Monthly_Salary, Annual_
33     val = (employee_id, name, department, job_title, experience_years, monthly_salary, annual_salary, total_earnings)
34     mycursor.execute(sql, val)
35     mydb.commit()
36
```

```
37 # To Print back The output.
38 output_text = (
39     f"Employee ID: {employee_id}\n"
40     f"Name: {name}\n"
41     f"Department: {department}\n"
42     f"Job Title: {job_title}\n"
43     f"Experience Years: {experience_years}\n"
44     f"Monthly Salary: RM{monthly_salary}\n"
45     f"Annual Salary: RM{annual_salary}\n"
46     f"Total Earnings: RM{total_earnings}"
47 )
48
49 output_label.config(text=output_text, foreground="blue", font=('Arial', 12, 'bold'))
50
51 # Function to update job titles based on the selected department
52 def update_job_titles(*args):
53     selected_department = department_var.get()
54     job_title_dropdown['values'] = job_titles.get(selected_department, [])
55
56 # Main window
57 root = tk.Tk()
58 root.title("Employee Management System")
59 root.geometry('600x700')
60 root.configure(bg='#FFECB3') # Set background color
61
62 # Style
63 style = ttk.Style()
64 style.configure('TLabel', font=('Arial', 14), foreground='#cc34ff') # Set label color
65 style.configure('TButton', font=('Arial', 13))
66
67 # Page Title
68 title_label = ttk.Label(root, text='Add an Employee', style='TLabel', background='#cdf3ff') # Set title background color
69 title_label.pack(pady=20)
70
```

```
71 # Employee ID Entry
72 id_label = ttk.Label(root, text="Employee ID:", background='#FFECB3')
73 id_label.pack()
74 entry_id = tk.Entry(root)
75 entry_id.pack()
76
77 # Employee Name Entry
78 name_label = ttk.Label(root, text="Name:", background='#FFECB3')
79 name_label.pack()
80 entry_name = tk.Entry(root)
81 entry_name.pack()
82
83 # Department Entry (Dropdown)
84 department_label = ttk.Label(root, text="Department:", background='#FFECB3')
85 department_label.pack()
86 departments = ["Executive", "Sales", "Marketing", "IT", "Operations", "Human Resources", "Finance"]
87 department_var = tk.StringVar(root)
88 department_var.set(departments[0]) # default value
89 department_var.trace_add('write', update_job_titles) # update job titles when department changes
90 department_dropdown = ttk.Combobox(root, textvariable=department_var, values=departments, state='readonly')
91 department_dropdown.pack()
92
```



```

93 # Job Title Entry (Dropdown)
94 job_title_label = ttk.Label(root, text="Job Title:", background='#FFECB3')
95 job_title_label.pack()
96 job_titles = {'Executive': ['CEO', 'CFO', 'CIO', 'CMO', 'COO'],
97               'Sales': ['Sales Manager', 'Sales Representative', 'Account Executive', 'Sales Analyst'],
98               'Marketing': ['Marketing Manager', 'Marketing Coordinator', 'Social Media Specialist', 'SEO Specialist'],
99               'IT': ['IT Manager', 'Systems Administrator', 'Network Engineer', 'Software Developer'],
100               'Operations': ['Operations Manager', 'Project Manager', 'Business Analyst', 'Quality Assurance Manager'],
101               'Human Resources': ['HR Manager', 'HR Generalist', 'Recruiter', 'Training and Development Manager'],
102               'Finance': ['Finance Manager', 'Accountant', 'Financial Analyst', 'Payroll Specialist']}
103 job_title_var = tk.StringVar(root)
104 job_title_var.set(job_titles['Executive'][0]) # default value
105 job_title_dropdown = ttk.Combobox(root, textvariable=job_title_var, values=job_titles['Executive'], state='readonly')
106 job_title_dropdown.pack()
107
108 # Experience Entry
109 experience_label = ttk.Label(root, text="Experience (in years):", background='#FFECB3')
110 experience_label.pack()
111 entry_experience = ttk.Entry(root)
112 entry_experience.pack()
113
114 # Salary Entry
115 salary_label = ttk.Label(root, text="Monthly Salary:", background='#FFECB3')
116 salary_label.pack()
117 entry_salary = ttk.Entry(root)
118 entry_salary.pack()
119
120 # Save Button
121 save_button = ttk.Button(root, text="Add Employee", command=add_employee, style='TButton')
122 save_button.pack(pady=15)
123
124 # Output Label & result
125 output_label = ttk.Label(root, text="Employee Details", font=('Arial', 12, 'bold'), background='#cdf3ff', foreground='#cc34ff')
126 output_label.pack(pady=10)
127
128 root.mainloop()
129

```

4.0 SNAPSHOT OF GUI



The screenshot displays a web application window titled "Employee Management System". The main content area has a light yellow background. At the top, there is a section header "Add an Employee" in purple text. Below this, the form consists of several labeled input fields: "Employee ID:" with a text box, "Name:" with a text box, "Department:" with a dropdown menu showing "Executive", "Job Title:" with a dropdown menu showing "CEO", "Experience (in years):" with a text box, and "Monthly Salary:" with a text box. A button labeled "Add Employee" is positioned below the salary field. At the bottom of the form area, there is another section header "Employee Details" in purple text.

Employee Management System

Add an Employee

Employee ID:

Name:

Department:

Executive

Job Title:

CEO

Experience (in years):

Monthly Salary:

Add Employee

Employee Details

5.0 SNAPSHOT OF DATABASE

The image displays three sequential screenshots of the phpMyAdmin interface, illustrating the process of viewing and managing a database table named 'employees' within the 'employee_management' database.

Top Screenshot: Database Overview

The top panel shows the 'Database: employee_management' overview. The 'Structure' tab is active, displaying a table named 'employees' with 7 rows and 8 columns. The columns are: Employee_ID, Employee_Name, Employee_Department, Job_Title, Experience_Years, Monthly_Salary, Annual_Salary, and Total_Earnings. The table is using the InnoDB engine and utf8mb4_general_ci collation.

Middle Screenshot: Data View

The middle panel shows the 'Table: employees' data view. The 'Browse' tab is active, displaying the data for the 'employees' table. The table has 7 rows and 8 columns. The data is as follows:

| Employee_ID | Employee_Name | Employee_Department | Job_Title | Experience_Years | Monthly_Salary | Annual_Salary | Total_Earnings |
|-------------|--------------------------------|---------------------|-------------------------|------------------|----------------|---------------|----------------|
| 2022060634 | NURUL ANIS ADEELA BINTI ROSELE | Executive | CEO | 4 | 291482.00 | 3497784.00 | 13991136.00 |
| 2022060612 | NURUL HANIN BINTI ROSLAN | Marketing | Social Media Specialist | 3 | 6789.00 | 81468.00 | 244404.00 |
| 2022060613 | NURUL HANIS BINTI OMAR | IT | Systems Administrator | 5 | 7890.00 | 94680.00 | 473400.00 |
| 2022060614 | AHMAD BIN DIN | Operations | Business Analyst | 6 | 6754.00 | 81048.00 | 486288.00 |
| 2022060615 | SULAIMAN BIN IDRIS | Human Resources | Recruiter | 7 | 6785.00 | 81420.00 | 569940.00 |
| 2022060616 | HAMZAH BIN ZAIN | Sales | Account Executive | 7 | 7865.00 | 94380.00 | 660660.00 |
| 2022060617 | ZULAIKHA BINTI ZULFAHMI | Finance | Accountant | 2 | 3456.00 | 41472.00 | 82944.00 |

Bottom Screenshot: Table Structure

The bottom panel shows the 'Table: employees' structure view. The 'Table structure' tab is active, displaying the table structure. The table has 8 columns: Employee_ID, Employee_Name, Employee_Department, Job_Title, Experience_Years, Monthly_Salary, Annual_Salary, and Total_Earnings. The columns are defined as follows:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---------------------|---------------|--------------------|------------|------|---------|----------|-------|------------------|
| 1 | Employee_ID | int(10) | | | No | None | | | Change Drop More |
| 2 | Employee_Name | varchar(100) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | Employee_Department | varchar(100) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | Job_Title | varchar(100) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | Experience_Years | int(100) | | | No | None | | | Change Drop More |
| 6 | Monthly_Salary | decimal(12,2) | | | No | None | | | Change Drop More |
| 7 | Annual_Salary | decimal(12,2) | | | No | None | | | Change Drop More |
| 8 | Total_Earnings | decimal(12,2) | | | No | None | | | Change Drop More |

6.0 CONCLUSION

In conclusion, the 'Employee Management System' is a user-friendly application developed with Python and Tkinter, aimed at simplifying the entry and management of critical personnel data. While currently not connected to a database, it offers colorful buttons, dropdown menus, and validation checks for accurate information. The system allows users to effortlessly add employees, select departments and job titles, and calculate essential details such as annual salary. Its adaptability makes it a valuable tool for maintaining organizational efficiency in handling employee data, with the potential for future database integration.