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

UNIVESITI 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

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

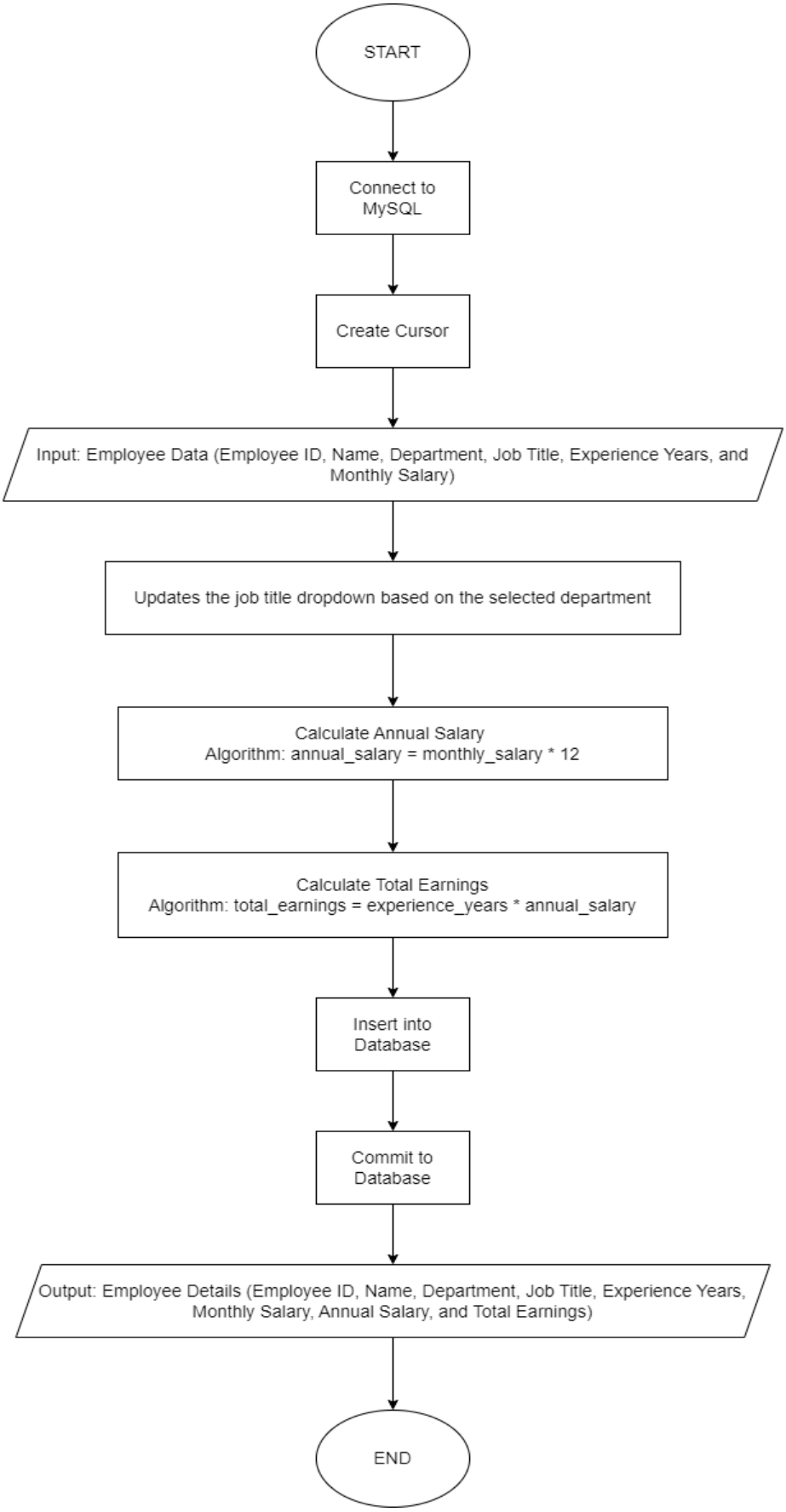


Figure 1 Flowchart Employee Management

3.0 SNAPSHOT OF CODE

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

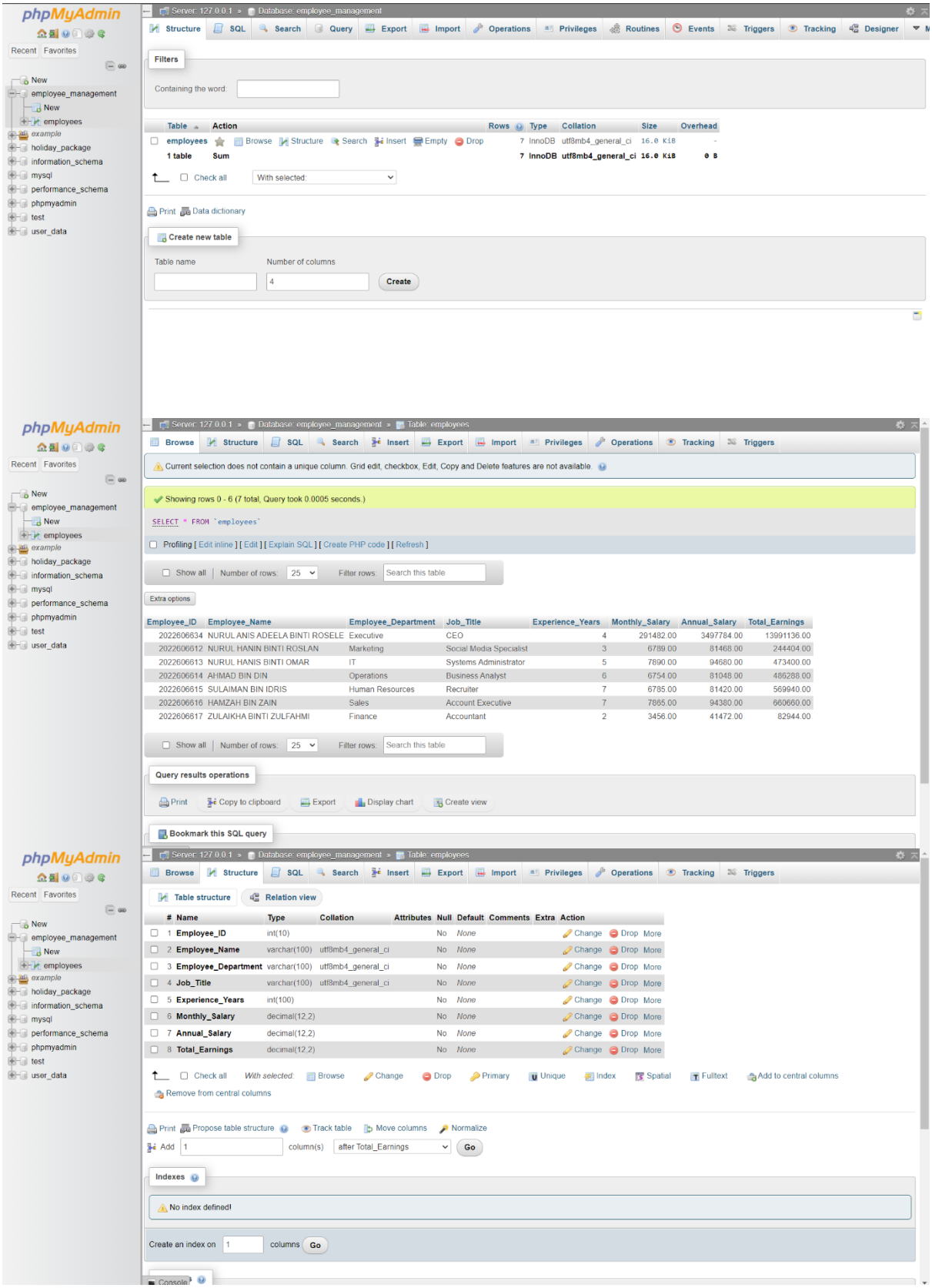
```python
import tkinter as tk
import tkinter.ttk as ttk
import mysql.connector

# Connect to your MySQL database
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="employee_management"
)

# Create a cursor object to execute SQL queries
mycursor = mydb.cursor()

# Function to handle the employee addition and database saving
def add_employee():
    employee_id = int(entry_id.get())
    name = entry_name.get()
    department = department_var.get()
    job_title = job_title_var.get()
    experience_years = int(entry_experience.get())
    monthly_salary = float(entry_salary.get())

    # Calculate annual salary
    annual_salary = monthly_salary * 12

    # Calculate total earnings based on experience years
    total_earnings = experience_years * annual_salary

    # To insert your Data into your database
    sql = "INSERT INTO `employees` (Employee_ID, Employee_Name, Employee_Department, Job_Title, Experience_Years, Monthly_Salary, Annual_
    val = (employee_id, name, department, job_title, experience_years, monthly_salary, annual_salary, total_earnings)
    mycursor.execute(sql, val)
    mydb.commit()

    # To Print back The output.
    output_text = (
        f"Employee ID: {employee_id}\n"
        f"Name: {name}\n"
        f"Department: {department}\n"
        f"Job Title: {job_title}\n"
        f"Experience Years: {experience_years}\n"
        f"Monthly Salary: RM{monthly_salary}\n"
        f"Annual Salary: RM{annual_salary}\n"
        f"Total Earnings: RM{total_earnings}"
    )

    output_label.config(text=output_text, foreground="blue", font=('Arial', 12, 'bold'))

# Function to update job titles based on the selected department
def update_job_titles(*args):
    selected_department = department_var.get()
    job_title_dropdown['values'] = job_titles.get(selected_department, [])

# Main window
root = tk.Tk()
root.title("Employee Management System")
root.geometry('600x700')
root.configure(bg='#FFECB3')  # Set background color

# Style
style = ttk.Style()
style.configure('TLabel', font=('Arial', 14), foreground='#cc34ff')  # Set label color
style.configure('TButton', font=('Arial', 13))

# Page Title
title_label = ttk.Label(root, text='Add an Employee', style='TLabel', background='#cdf3ff')  # Set title background color
title_label.pack(pady=20)

# Employee ID Entry
id_label = ttk.Label(root, text="Employee ID:", background='#FFECB3')
id_label.pack()
entry_id = ttk.Entry(root)
entry_id.pack()

# Employee Name Entry
name_label = ttk.Label(root, text="Name:", background='#FFECB3')
name_label.pack()
entry_name = ttk.Entry(root)
entry_name.pack()

# Department Entry (Dropdown)
department_label = ttk.Label(root, text="Department:", background='#FFECB3')
department_label.pack()
departments = ["Executive", "Sales", "Marketing", "IT", "Operations", "Human Resources", "Finance"]
department_var = tk.StringVar(root)
department_var.set(departments[0])  # default value
department_var.trace_add('write', update_job_titles)  # update job titles when department changes
department_dropdown = ttk.Combobox(root, textvariable=department_var, values=departments, state='readonly')
department_dropdown.pack()

```

```
# Job Title Entry (Dropdown)
job_title_label = ttk.Label(root, text="Job Title:", background='#FFECB3')
job_title_label.pack()
job_titles = {'Executive': ['CEO', 'CFO', 'CIO', 'CMO', 'COO'],
              'Sales': ['Sales Manager', 'Sales Representative', 'Account Executive', 'Sales Analyst'],
              'Marketing': ['Marketing Manager', 'Marketing Coordinator', 'Social Media Specialist', 'SEO Specialist'],
              'IT': ['IT Manager', 'Systems Administrator', 'Network Engineer', 'Software Developer'],
              'Operations': ['Operations Manager', 'Project Manager', 'Business Analyst', 'Quality Assurance Manager'],
              'Human Resources': ['HR Manager', 'HR Generalist', 'Recruiter', 'Training and Development Manager'],
              'Finance': ['Finance Manager', 'Accountant', 'Financial Analyst', 'Payroll Specialist']}
job_title_var = tk.StringVar(root)
job_title_var.set(job_titles['Executive'][0])  # default value
job_title_dropdown = ttk.Combobox(root, textvariable=job_title_var, values=job_titles['Executive'], state='readonly')
job_title_dropdown.pack()

# Experience Entry
experience_label = ttk.Label(root, text="Experience (in years):", background='#FFECB3')
experience_label.pack()
entry_experience = ttk.Entry(root)
entry_experience.pack()

# Salary Entry
salary_label = ttk.Label(root, text="Monthly Salary:", background='#FFECB3')
salary_label.pack()
entry_salary = ttk.Entry(root)
entry_salary.pack()

```

```
# Save Button
save_button = ttk.Button(root, text="Add Employee", command=add_employee, style='TButton')
save_button.pack(pady=15)

# Output Label & result
output_label = ttk.Label(root, text="Employee Details", font=('Arial', 12, 'bold'), background='#cdf3ff', foreground='#cc34ff')
output_label.pack(pady=10)

root.mainloop()

```

4.0 SNAPSHOT OF GUI

5.0 SNAPSHOT OF DATABASE



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.