BSD2213 DATA SCIENCE PROGRAMMING I

FINANCIAL ASSISTANCE APPLICATION

LECTURER'S NAME: DR NORAZIAH BINTI ADZHAR

MATRIC ID	NAME
SD22059	LOW ANN GIE
SD22060	FARAH HANAN BINTI TAJUDIN
SD22025	ABDUL HAZIQ AZIM BIN ABDUL MALIK
SD22044	NUR AZALEA ZAFINA BINTI MOHD FAIZAL

TABLE OF CONTENTS

No	Contents	Pages
1	Introduction	3
2	Why this project?	4-5
3	Extension Ideas on Project	6-7
4	Source Code	8-19
5	Screenshot of activity in GUI	20-27

INTRODUCTION

This project is to create a user-friendly financial application that addresses the specific needs of the low-income community, promoting financial inclusion and empowerment. In a low-income community, they are struggling with economic hardship and may leading to difficulties in meeting basic needs such as housing, healthcare and education. The existing of social support systems may not be sufficient to address the diverse financial needs of the residents.

There are some reasons might cause the residents having financial problems. Firstly, the community may have high unemployment rates that make it challenging for individuals to secure stable sources of income. Next, the limited access to financial resources may cause the residents may lack access to traditional banking services, credit facilities or affordable loans, making it difficult for them to overcome financial obstacles. Lastly, the technological divide also is one of the reasons why some individuals may face challenges in accessing a navigating online platform due to a lack of digital literacy or limited access to technology. By addressing the challenges associated with the technological divide, the financial assistance application project can strive for inclusivity ensuring that all members of the community have an equal opportunity to access and benefit from available financial assistance services.

This financial assistance application incorporates a range of key features designed to address the diverse needs of a low-income community. A user-friendly interface has been prioritized, ensuring accessibility for people of all digital literacy levels. The project's goal is to bridge existing gaps in social support systems and empower individuals to overcome economic hardships effectively by developing a financial assistance application tailored to the unique challenges of the low-income community. The success of this project would be measured not only by the number of people who received assistance but also by the long-term impact on the community's economic resilience and self-sufficiency.

WHY THIS PROJECT?

Financial assistance applications play a crucial role in providing support to individuals and businesses facing economic challenges. In recent years, the adoption of digital platforms for financial assistance has become increasingly prevalent. This report outlines the significance of financial assistance applications, emphasizing their impact on accessibility, efficiency, transparency, and overall effectiveness in addressing financial needs.

The financial assistance applications can remove issues to accessibility by providing a constant presence. This is a significant advantage compared to older techniques that frequently limit accessibility due to office hours or geographical constraints. Applications and information are accessible to users all the time, ensuring that financial support is available when it's most needed.

Applications for financial assistance allow easier processes which greatly increase their effectiveness. Workflows that are automated lighten the administrative burden and increase application and funding processing. This effectiveness helps applicants by giving them timely support. It also relieves pressure on organisations in the position of managing financial aid programmes. The instantaneous updates regarding the status of applications provide applicants with further efficiency increases.

Transparency is essential to providing efficient financial support, and applications have a crucial role to making this happen. Trackable transactions can be found on digital platforms, giving an open record of the money's movement. This encourages user trust while ensuring accountability. Furthermore, strict data security protocols secure personal financial data, giving applicants peace of mind about the security of their personal information.

One important benefit of applications for financial assistance is their flexibility, which makes it possible to modify assistance plans to meet specific demands or needs. This adaptation guarantees that support is focused and relevant to each person's situation. Interfaces that are easy to use improve the applicant experience by simplifying the application process and facilitating the submission of required materials.

Applications for financial help are the catalyst for change in the field of economic support. Their substantial impact is mostly due to the benefits of improved accessibility, efficiency, transparency, cost-effectiveness, and modification. Financial assistance platforms have a great deal of room for innovation as technology develops, which might lead to better support and responsiveness to the constant needs of people with limited resources and enterprises. Financial help will surely change in the future because of the ongoing advancement of technology and careful application design, providing an offer of stability and hope for people who are struggling financially.

EXTENSIONS IDEAS ON THIS PROJECT

In the relentless pursuit of progress and completion, the concept of improvement stands as a beacon guiding us towards more advancement with enhanced efficiency and perfection. As we are completing this project, we are aware of its capability and potential to be more impactful. This Financial Assistance Application project can be extended by:

1. In-App Chat Support

In-app Chat support has become a vital component in modern customer service. This feature will allow users to connect with customer support anywhere in real time. This dynamic tool will also guide users of all ages and provide instant solutions swiftly.

2. More thorough screening and application process

Effective screening and more detailed thorough application are needed to ensure the most qualified applicant is accepted. By implementing more detailed screening, we can delve deeper into application qualifications, certifications and financial acumen. Therefore, the most deserving applicants will pass without any issues.

3. Interactable widgets and features

Interactable widget is a pivotal component in user interface design and can enrich user experience by offering dynamic and responsive elements. These widgets are designed to engage users actively and manipulate content. For example, a statistics widget that shows the number of applicants passed through the years. With these, the user experience will overall be more usable and enjoyable.

4. More category for applicants

Incorporating additional categories and levels can bolster its effectiveness and inclusivity. By expanding the range of categories, the app can better cater to diverse financial needs, ensuring a more nuanced and tailored approach for users. Introducing different levels of assistance enables a finer granularity in addressing the varying degrees of financial challenges individuals may face. It not only offers a more customised experience but also makes it possible to allocate resources more precisely.

SOURCE CODE

```
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
from PIL import Image, ImageTk

class FirstPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)

        self.config(bg='#fff')

        self.img = PhotoImage(file='login.png')
        Label(self, image=self.img, bg='white').place(x=80,
y=80)

        frame = Frame(self, width=350, height=350, bg='white')
        frame.place(x=650, y=130)

        heading = Label(frame, text='Sign in', fg='#57a1f8',
bg='white', font=('Microsoft YaHei UI Light', 23, 'bold'))
        heading.place(x=115, y=10)

        title = Label(self, text='Financial Assistance
Application', bg='white', font=('Arial', 30, 'bold'))
        title.place(x=500, y=38)



        def on_enter(e):
            user.delete(0, 'end')

        def on_leave(e):
            name = user.get()
            if name == '':
                user.insert(0, 'IC Number')

        user = Entry(frame, width=25, fg='black', border=0,
bg='white', font=('Microsoft YaHei UI Light', 11))
        user.place(x=30, y=80)
        user.insert(0, 'IC Number')
        user.bind('<FocusIn>', on_enter)
        user.bind('<FocusOut>', on_leave)

        Frame(frame, width=295, height=2,
bg='black').place(x=25, y=107)
```

```
        def on_enter(e):
            username.delete(0, 'end')

        def on_leave(e):
            name = username.get()
            if name == '':
                username.insert(0, 'Username')

        username = Entry(frame, width=25, fg='black',
border=0, bg='white', font=('Microsoft YaHei UI Light', 11))
        username.place(x=30, y=150)
        username.insert(0, 'Username')
        username.bind('<FocusIn>', on_enter)
        username.bind('<FocusOut>', on_leave)

        Frame(frame, width=295, height=2,
bg='black').place(x=25, y=177)


        def on_enter(e):
            code.delete(0, 'end')

        def on_leave(e):
            name = code.get()
            if name == '':
                code.insert(0, 'Password')

        code = Entry(frame, width=25, fg='black', border=0,
bg='white', font=('Microsoft YaHei UI Light', 11))
        code.place(x=30, y=220)
        code.insert(0, 'Password')
        code.bind('<FocusIn>', on_enter)
        code.bind('<FocusOut>', on_leave)

        Frame(frame, width=295, height=2,
bg='black').place(x=25, y=247)



        def verify():

                try:
                    with open('credential.txt', 'r') as f:
                        info = f.readlines()
                        i = 0
                        for e in info:
                            u, p = e.split(',')
                            if u.strip() == user.get() and
p.strip() == code.get():
                                self.user_info = {
                                    'username':
```

```
username.get(),
                                        'ic_number': user.get()}

controller.show_frame(FourthPage, self.user_info)
                                    i = 1
                                    break
                            if i == 0:
                                messagebox.showinfo('Error',
'Please provide correct IC Number and password!')
                    except:
                        messagebox.showinfo('Error', 'Please
provide correct IC Number and password!')



        Button(frame, width=39, pady=7, text='Sign in',
bg='#57a1f8', fg='white', border=0,
command=verify).place(x=35, y=267)
        label = Label(frame, text="Don't have an account?",
fg='black', bg='white',font=('Microsoft YaHei UI Light', 9))
        label.place(x=75, y=333)

        def register():
            controller.show_frame(SecondPage)

        sign_up = Button(frame, width=6, text='Sign up',
border=0, bg='white', fg='#57a1f8', command=register)
        sign_up.place(x=215, y=333)



class SecondPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)

        self.config(bg='white')

        img = Image.open('registration.png')
        photo = ImageTk.PhotoImage(img)
        label = Label(self, image=photo)
        label.image=photo
        label.place(x=0, y=0)

        frame = Frame(self, width=950, height=400, bg='white')
        frame.place(x=125, y=70)

        label = Label(frame, text = 'Registration',
bg='white', font=('Arial', 30, 'bold'))
        label.place(x=360, y=30)
```

```
        l1 = Label(frame, text='IC Number', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l1.place(x=250, y=150)
        user = Entry(frame, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        user.place(x=500, y=150)

        l2 = Label(frame, text='Password', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l2.place(x=250, y=210)
        code = Entry(frame, width=25, fg='black', bd=3,
show='*',bg='white', font=('Microsoft YaHei UI Light', 11))
        code.place(x=500, y=210)

        l3 = Label(frame, text='Confirm Password', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l3.place(x=250, y=270)
        confirm = Entry(frame, width=25, fg='black', show='*',
bd=3, bg='white', font=('Microsoft YaHei UI Light', 11))
        confirm.place(x=500, y=270)

        accept = StringVar(value="Not Accepted")
        l4 = Checkbutton(frame, text='I agree to the Term &
Conditions', font=('Microsoft YaHei UI Light', 9), bg='white',
variable=accept, onvalue="Accepted", offvalue="Not Accepted")
        l4.place(x=385, y=310)



        def check():
            if user.get() != '' and code.get() != '' and
confirm.get() != '':
                if code.get() == confirm.get():
                    with open('credential.txt', 'a') as f:
                        f.write(user.get() + ',' + code.get()
+ '\n')
                    if accept.get() == "Accepted":
                        messagebox.showinfo('Welcome', 'you
are registered successfully!!')
                        controller.show_frame(ThirdPage)
                    else:
                        messagebox.showinfo('Error', "Please
accept the Term & Conditions!!")

                else:
                    messagebox.showinfo('Error', "Your
password didn't get match!!")
            else:
                messagebox.showinfo('Error', "Please fill the
complete field!!")
```

```
        Button(frame, width=39, pady=7, text='Sign up',
bg='#57a1f8', fg='white', border=0,
command=check).place(x=360,y=350)



class ThirdPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)

        self.config(bg='white')

        label = Label(self, text='Information Application',
bg='white', font=('Arial', 30, 'bold'))
        label.place(x=400, y=15)

        f1 = Frame(self, width=570, height=450, bd=2,
bg='white')
        f1.place(x=15, y=75)

        f2 = Frame(self, width=570, height=420, bd=2,
bg='white')
        f2.place(x=615, y=75)

        l1 = Label(f1, text='IC Number', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l1.place(x=30, y=25)
        e1 = Entry(f1, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e1.place(x=250, y=25)

        l2 = Label(f1, text='Name', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l2.place(x=30, y=85)
        e2 = Entry(f1, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e2.place(x=250, y=85)

        l3 = Label(f1, text='Job', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l3.place(x=30, y=145)
        e3 = Entry(f1, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e3.place(x=250, y=145)

        l4 = Label(f1, text='Allowance', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l4.place(x=30, y=205)
        e4 = Entry(f1, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e4.place(x=250, y=205)
```

```
        l5 = Label(f1, text='Address', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l5.place(x=30, y=265)
        e5 = Entry(f1, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e5.place(x=250, y=265)

        l6 = Label(f1, text='Bank Name', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l6.place(x=30, y=325)
        Bank = ['BANK SIMPANAN NASIONAL', 'AFFIN BANK BERHAD',
'ARGO BANK', 'AL RAJHI BANKING INVESTMENT CORPORATION',
'ALLIANCE BANK MALAYSIA BERHAD', 'AM BANK BERHAD', 'BANK ISLAM
MALAYSIA BERHAD', 'BANK KERJASAMA RAKYAT MALAYSIA BHD', 'BANK
MUAMALAT MALAYSIA BERHAD', 'BANK OF CHINA (MALAYSIA) BERHAD',
'CIMB BANK BERHAD', 'CITIBANK BERHAD', 'HONG LEONG BANK',
'HONGKONG SHANGHAI BANK MALAYSIA BHD', 'KUWAIT FINANCE HOUSE
MALAYSIA', 'MALAYAN BANKING BERHAD', 'OCBC AL-AMIN BANK
BHD/OCBC BANK MALAYSIA BHD', 'PUBLIC BANK BERHAD/PUBLIC
ISLAMIC BANK BERHAD', 'RHB BANK BERHAD']
        e6 = ttk.Combobox(f1, values=Bank, width=31)
        e6.place(x=250, y=325)

        l7 = Label(f1, text='Bank Account Number', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l7.place(x=30, y=385)
        e7 = Entry(f1, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e7.place(x=250, y=385)



        l8 = Label(f2, text='Age', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l8.place(x=30, y=25)
        e8 = Entry(f2, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e8.place(x=250, y=25)

        l9 = Label(f2, text='Gender', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l9.place(x=30, y=85)
        v=StringVar()
        e9 = Radiobutton(f2, text='Male', variable=v,
value='Male', bg='white')
        e9.place(x=250, y=85)
        e9 = Radiobutton(f2, text='Female', variable=v,
value='Female', bg='white')
        e9.place(x=335, y=85)
```

```
        l10 = Label(f2, text='Marial Status', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l10.place(x=30, y=145)
        marial = ['single', 'married', 'divorced', 'widowed']
        e10 = ttk.Combobox(f2, values=marial, width=31)
        e10.place(x=250, y=145)

        l11 = Label(f2, text='Number of children', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l11.place(x=30, y=205)
        e11 = Spinbox(f2, from_=0, to=15, width=32)
        e11.place(x=250, y=205)

        l12 = Label(f2, text='Contact Number', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l12.place(x=30, y=265)
        e12 = Entry(f2, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e12.place(x=250, y=265)

        l13 = Label(f2, text='Email Address', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l13.place(x=30, y=325)
        e13 = Entry(f2, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        e13.place(x=250, y=325)


        def nextPage():

            if e1.get() != '' and e2.get() != '' and
e3.get() != '' and e4.get() != '' and e5.get() != '' and
e6.get() != '' and e7.get() != '' and e8.get() != '' and
v.get() != '' and e10.get() != '' and e11.get() != '' and
e12.get() != '' and e13.get() != '':
                self.user_info = {
                    'username': e2.get(),
                    'ic_number': e1.get()}
                with open('data.txt', 'a') as f:
                    f.write(e1.get() + ',' + e2.get() + ',' +
e3.get() + ',' + e4.get() + ',' + e5.get() + ',' + e6.get() +
',' + e7.get() + ',' + e8.get() + ',' + v.get() + ',' +
e10.get() + ',' + e11.get() + ',' + e12.get() + ',' +
e13.get() + '\n')
                messagebox.showinfo('Welcome', "Submit
successfully!!")
                controller.show_frame(FourthPage,
self.user_info)
            else:
                messagebox.showinfo('Error', "Please fill the
```

```
complete field!!")

        Button(self, width=20, pady=8, text='Next',
bg='#57a1f8', fg='white', border=0,
command=nextPage).place(x=1000,y=500)



class FourthPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)

        self.config(bg='white')

        try:
            f1 = Frame(self, width=370, height=350, bd=2,
bg='white')
            f1.place(x=15, y=100)

            f2 = Frame(self, width=370, height=350, bd=2,
bg='white')
            f2.place(x=415, y=100)

            f3= Frame(self, width=370, height=350, bd=2,
bg='white')
            f3.place(x=815, y=100)

            self.img1 = PhotoImage(file='update.png')
            Label(f1, image=self.img1, bg='white').place(x=70,
y=20)

            self.img2 = PhotoImage(file='result.png')
            Label(f2, image=self.img2, bg='white').place(x=70,
y=20)

            self.img3 = PhotoImage(file='change password.png')
            Label(f3, image=self.img3, bg='white').place(x=70,
y=25)

            def result():
                messagebox.showinfo('Status', "Application
Accepted, the application will be processed in March 2024")

            def changepass():
                controller.show_frame(FifthPage)

            b1 = Button(f1, text='Edit Application
Information', width=25, bg='yellow', font=('Arial', 15,
'bold'), command=lambda: controller.show_frame(ThirdPage))
            b1.place(x=27, y=290)
```

```
            b2 = Button(f2, text='Result for Application',
width=25, bg='yellow', font=('Arial', 15, 'bold'),
command=result)
            b2.place(x=27, y=290)

            b3 = Button(f3, text='Change Password', width=25,
bg='yellow', font=('Arial', 15, 'bold'), command=changepass)
            b3.place(x=27, y=290)

            def togglemenu():

                def collapse():
                    toggle_menu.destroy()
                    toggle_btn.config(text='☰')
                    toggle_btn.config(command=togglemenu)

                toggle_menu = Frame(self, bg='orange')

                home_button=Button(toggle_menu, text='Home',
font=('bold', 20),  bg='orange', fg='white', bd=0,
activebackground='orange', activeforeground='white',
command=lambda: controller.show_frame(FirstPage))
                home_button.place(x=20, y=20)

                def on_closing():
                    if messagebox.askyesno(title='Quit?',
message='Do you really want quit?'):
                        self.quit()
                        self.destroy()

                logOut_button = Button(toggle_menu, text='Log
Out', font=('bold', 20), bg='orange', fg='white', bd=0,

activebackground='orange', activeforeground='white',
command=on_closing)
                logOut_button.place(x=20, y=80)

                window_height = self.winfo_height()
                toggle_menu.place(x=1000, y=50, height=150,
width=200)

                toggle_btn.config(text='X')
                toggle_btn.config(command=collapse)

            head_frame =Frame(self, bg='orange',
highlightbackground='white', highlightthickness=1)

            title = Label(head_frame, text='Financial
Assistance Application', bg='orange', fg='white',
font=('bold', 20))
```

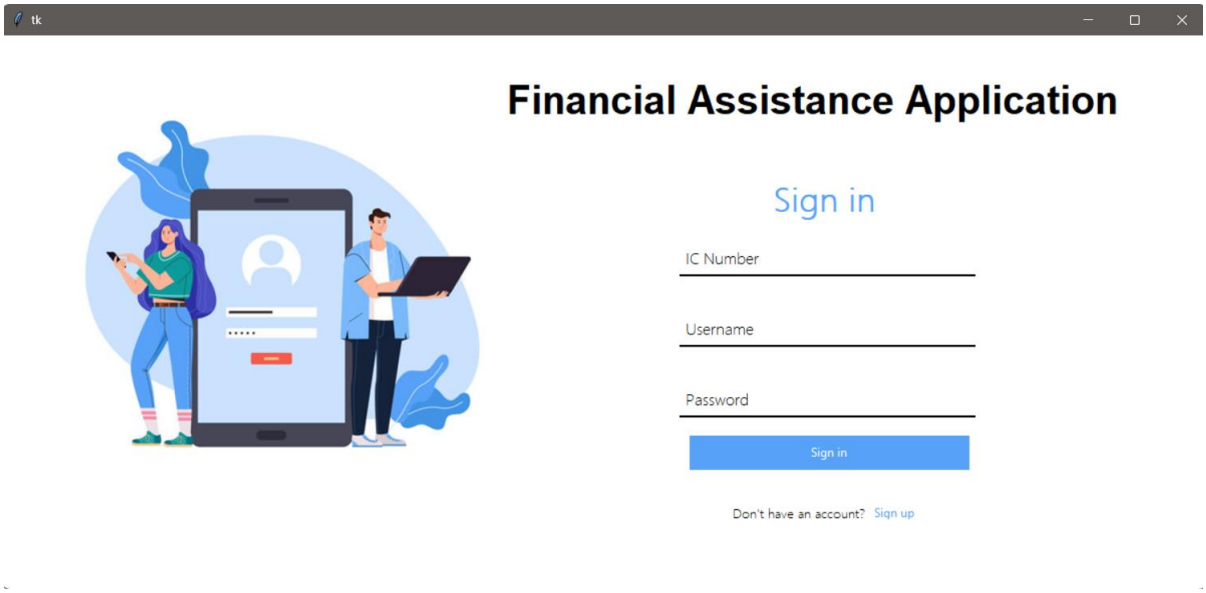
```
            title.pack(side=RIGHT)

            toggle_btn = Button(head_frame, text='☰',
bg='orange', fg='white', font=('bold', 20), bd=0,
activebackground='orange', activeforeground='white',
command=togglemenu)
            toggle_btn.pack(side=RIGHT)

            head_frame.pack(side=TOP, fill=X)
            head_frame.pack_propagate(False)
            head_frame.configure(height=50)


        except Exception as e:
            print(f"Error loading images: {e}")

        self.label_username = Label(self, text="Username:",
bg='white', font=('Arial', 16))
        self.label_username.pack(pady=5, anchor='nw')

        self.label_ic_number = Label(self, text="IC Number:",
bg='white', font=('Arial', 16))
        self.label_ic_number.pack(pady=10, anchor='nw')
    def on_show(self, user_info=None):

        if user_info is not None:
            self.label_username.config(text=f"Username:
{user_info.get('username', '')}")
            self.label_ic_number.config(text=f"IC Number:
{user_info.get('ic_number', '')}")
        else:
            self.label_username.config(text="Username: ")
            self.label_ic_number.config(text="IC Number: ")


class FifthPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)

        self.config(bg='white')

        img = Image.open('registration.png')
        photo = ImageTk.PhotoImage(img)
        label = Label(self, image=photo)
        label.image=photo
        label.place(x=0, y=0)

        frame = Frame(self, width=950, height=420, bg='white')
        frame.place(x=125, y=50)
```

```
        label = Label(frame, text = 'Change Password',
bg='white', font=('Arial', 30, 'bold'))
        label.place(x=320, y=20)

        l1 = Label(frame, text='IC Number', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l1.place(x=250, y=130)
        user = Entry(frame, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        user.place(x=500, y=130)

        l2 = Label(frame, text='Username', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l2.place(x=250, y=190)
        username = Entry(frame, width=25, bd=3, bg='white',
font=('Microsoft YaHei UI Light', 11))
        username.place(x=500, y=190)

        l3 = Label(frame, text='New Password', bg='white',
font=('Microsoft YaHei UI Light', 11, 'bold'))
        l3.place(x=250, y=250)
        code = Entry(frame, width=25, fg='black', bd=3,
show='*',bg='white', font=('Microsoft YaHei UI Light', 11))
        code.place(x=500, y=250)

        l4 = Label(frame, text='Confirm New Password',
bg='white', font=('Microsoft YaHei UI Light', 11, 'bold'))
        l4.place(x=250, y=310)
        confirm = Entry(frame, width=25, fg='black', show='*',
bd=3, bg='white', font=('Microsoft YaHei UI Light', 11))
        confirm.place(x=500, y=310)

        ##############################################--------
-------------------------------------------------------------
-------

        def check():
            if user.get() != '' and username.get() != ''and
code.get() != '' and confirm.get() != '':
                if code.get() == confirm.get():
                    self.user_info = {
                        'username': username.get(),
                        'ic_number': user.get()}
                    with open('credential.txt', 'a') as f:
                        f.write(user.get() + ',' + code.get()
+ '\n')
                        messagebox.showinfo('Welcome', 'your
password is changed successfully!!')
                    controller.show_frame(FourthPage,
self.user_info)
                else:
```

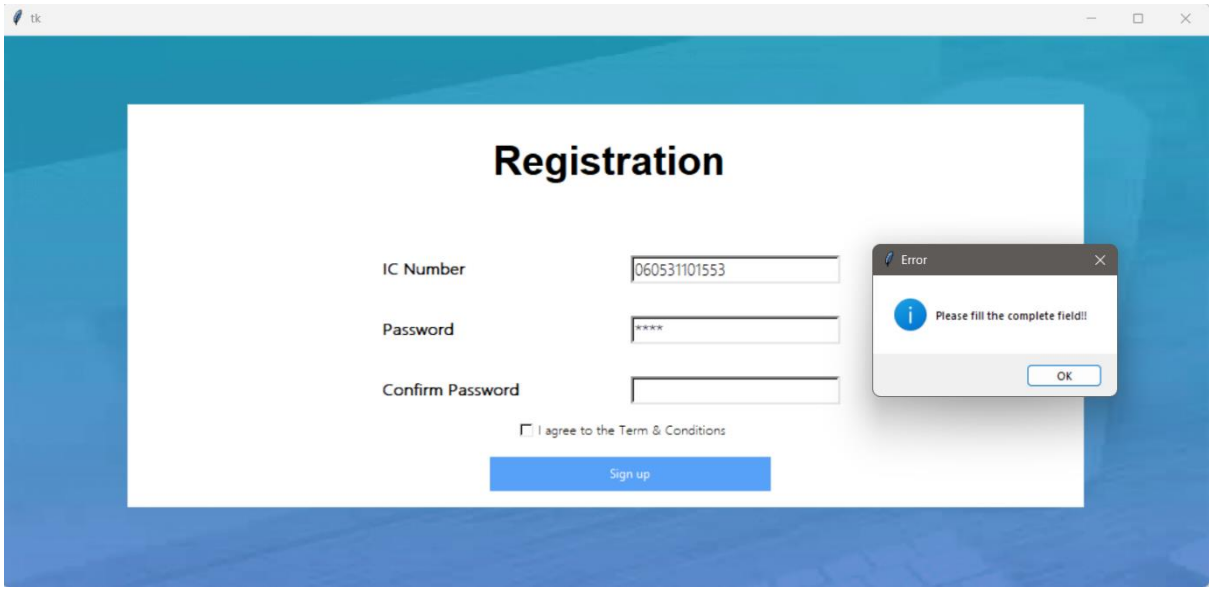
```
                    messagebox.showinfo('Error', "Your
password didn't get match!!")
            else:
                messagebox.showinfo('Error', "Please fill the
complete field!!")


        Button(frame, width=39, pady=7, text='Change
Password', bg='#57a1f8', fg='white', border=0,
command=check).place(x=350,y=375)


class Application(Tk):
    def __init__(self, *args, **kwargs):
        Tk.__init__(self, *args, **kwargs)


        window = Frame(self)
        window.pack()

        window.grid_rowconfigure(0, minsize = 550)
        window.grid_columnconfigure(0, minsize=1200)

        self.frames = {}
        for F in (FirstPage, SecondPage, ThirdPage,
FourthPage, FifthPage):
            frame = F(window, self)
            self.frames[F] = frame
            frame.grid(row = 0, column = 0, sticky='nsew')

        self.show_frame(FirstPage)

    def show_frame(self, page, *args, **kwargs):
        frame = self.frames[page]
        if hasattr(frame, 'on_show'):
            frame.on_show(*args, **kwargs)
        frame.tkraise()

app = Application()
app.maxsize(1200,550)
app.mainloop()
```

SCREENSHOT OF ACTIVITY IN GUI

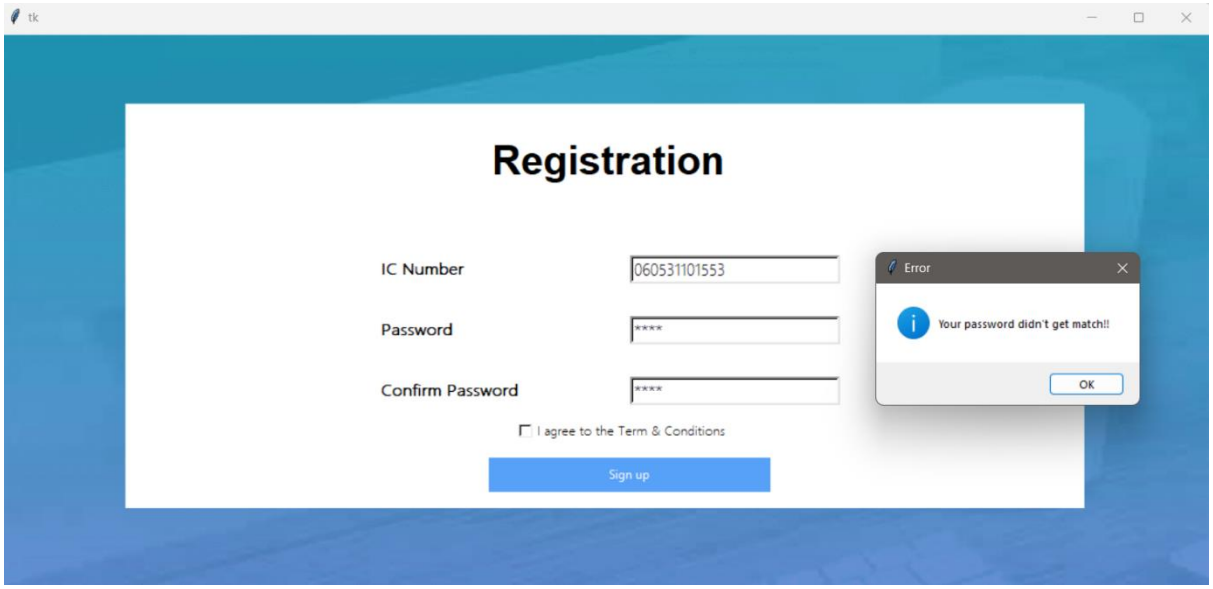
Page 1: Login Page. After pressing the Sign up, it will move to page 2.



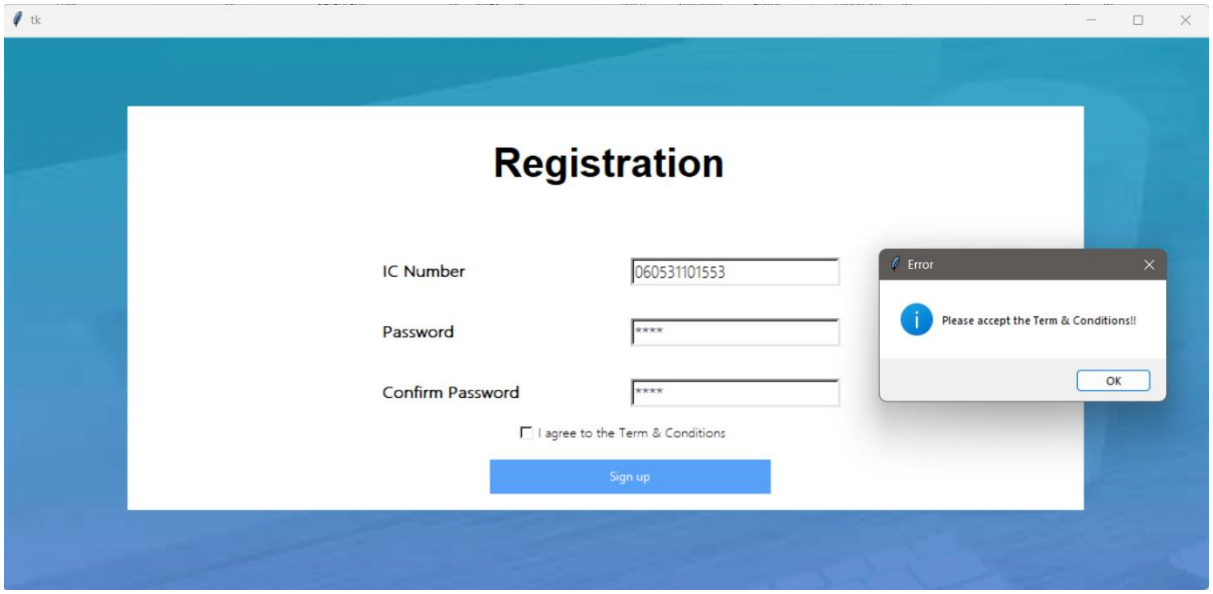
Page 2: If the user does not fill all the information, pressing Sign up button will show the message box with text ‘Please fill the complete field!!’.



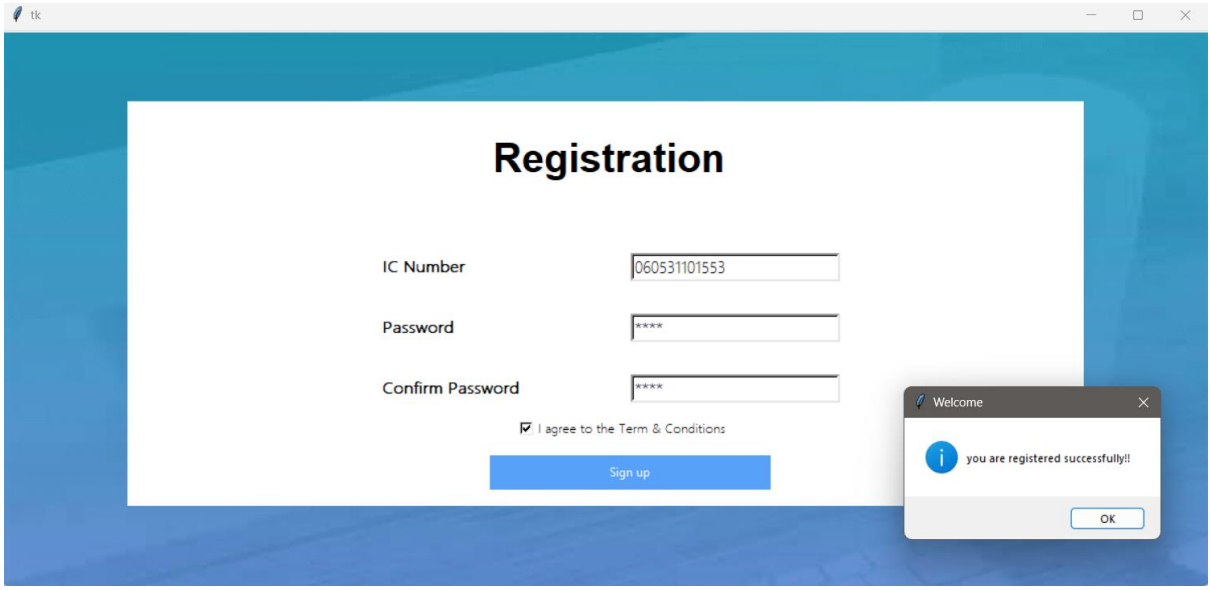
Page 2: If the user wrongly keys in the confirm password, pressing the ‘Sign up’ button will show the message box with the text ‘Your password didn’t get match!!’.



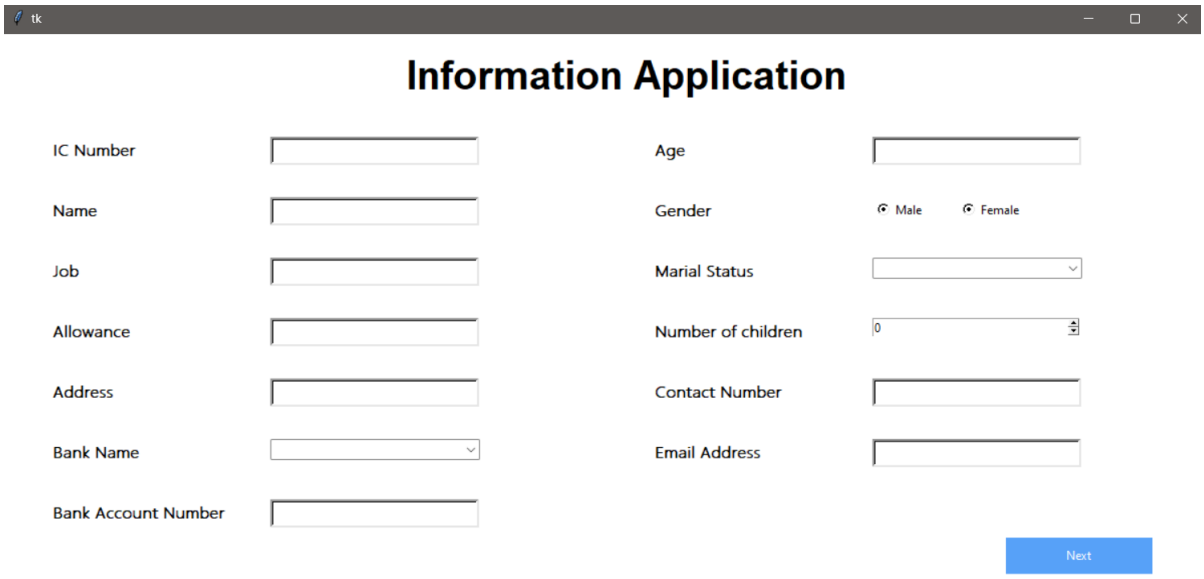
Page 2: If the user does not tick agree to the Term & Conditions, pressing sign up button will show the message box with the text ‘Please accept the Term & Conditions!!’.



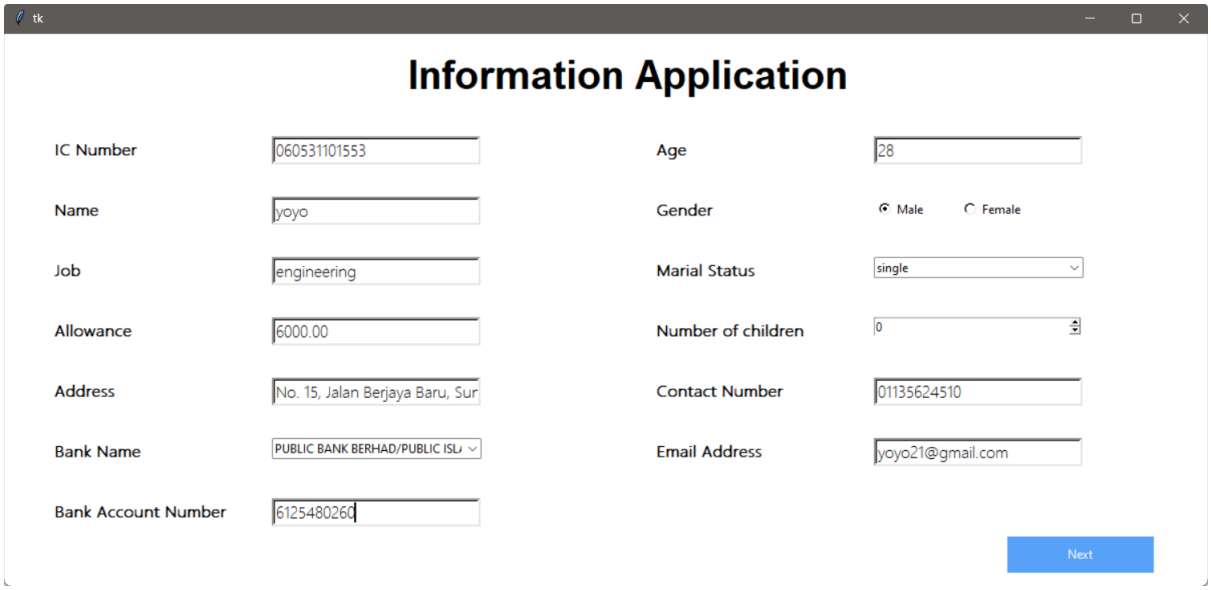
Page 2: After pressing the Sign up button, it will show the message box with the text ‘you are registered successfully!!’. It will turn to Page 3.



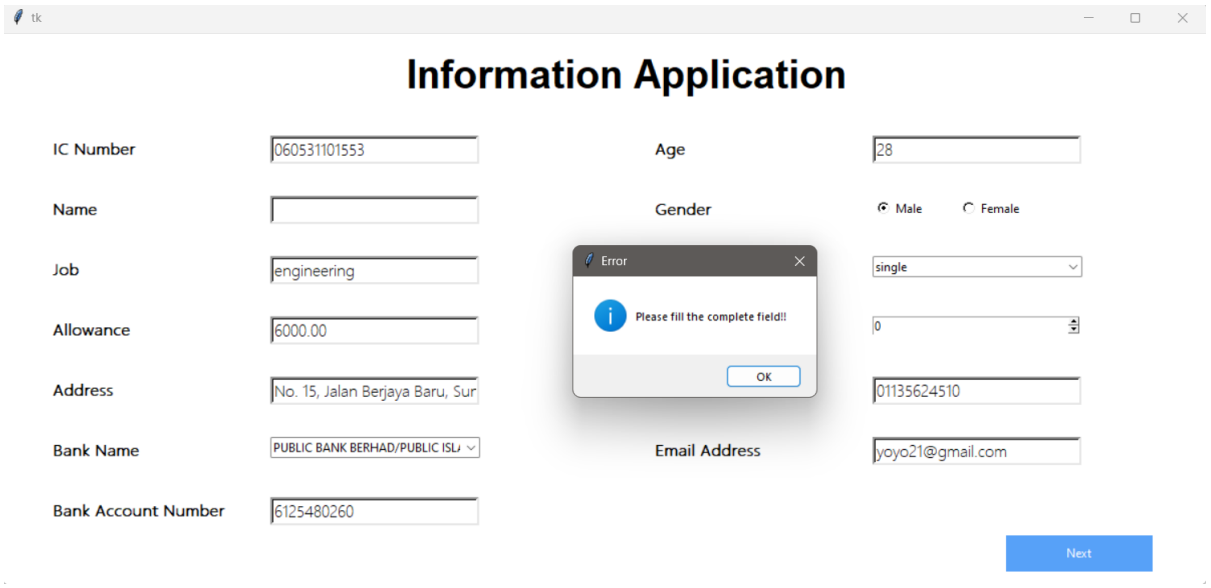
Page 3: Information Application.



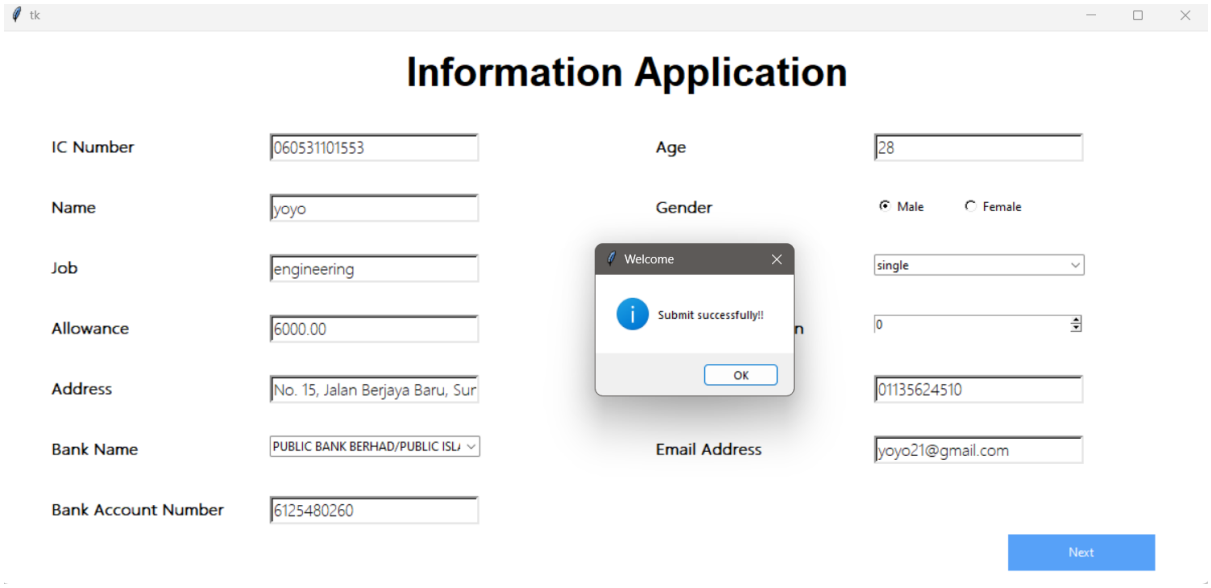
Page 3: Fill in the information for application.



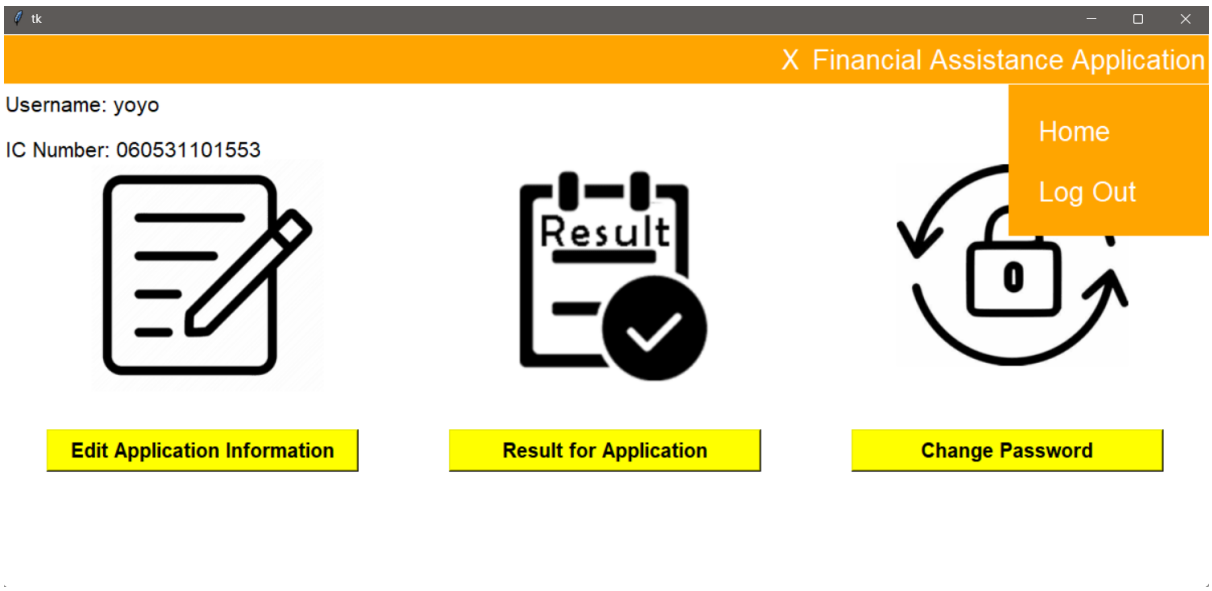
Page 3: If the user does not fill in all the information, pressing sign up button will show the message box with the text ‘Please fill the complete field!!’.



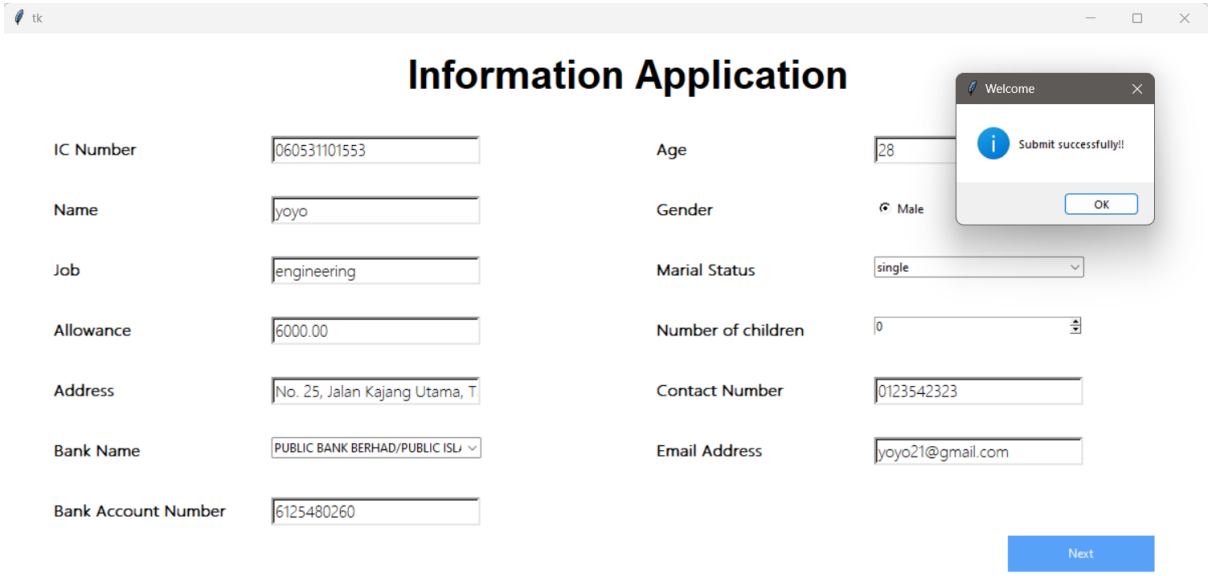
Page 3: After successfully submit the application, it will turn to page 4.



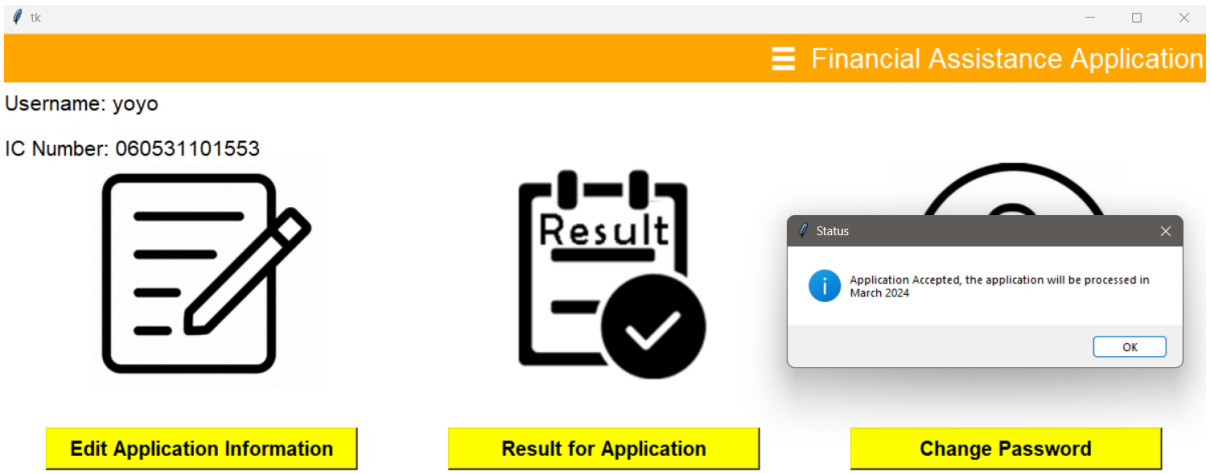
Page 4: Username and IC Number are displayed.



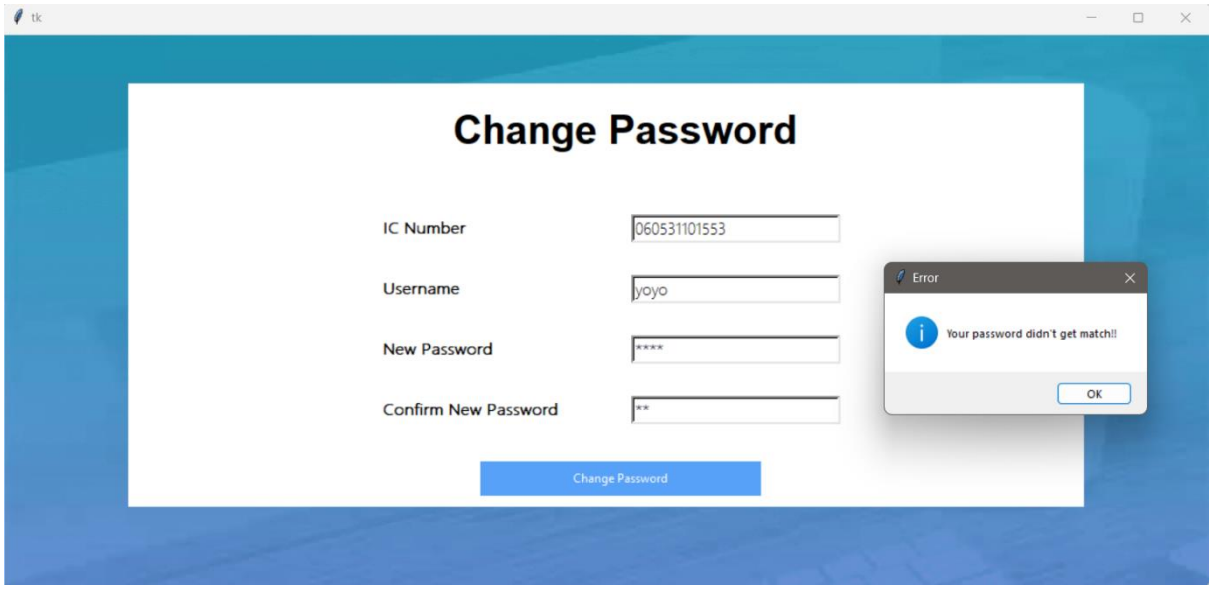
Page 4: After pressing Edit Application Information button, it will turn to this page. After successfully update the information, it will return to page 4.



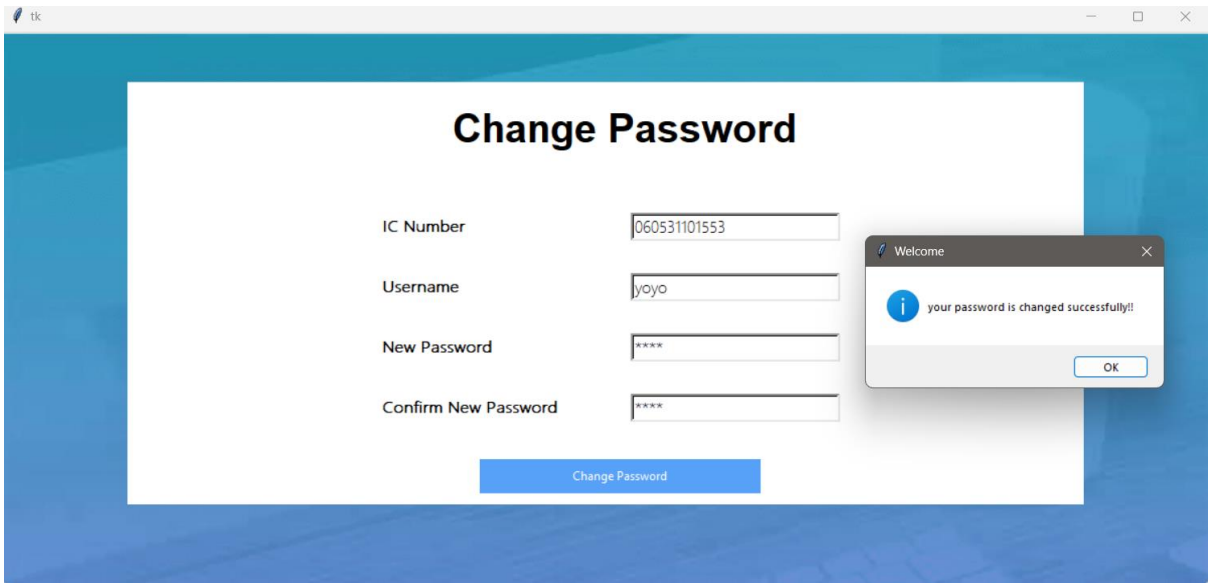
Page 4: After pressing Result for Application button, it show message box with the text 'Application Accepted, the application will be processed in March 2024'



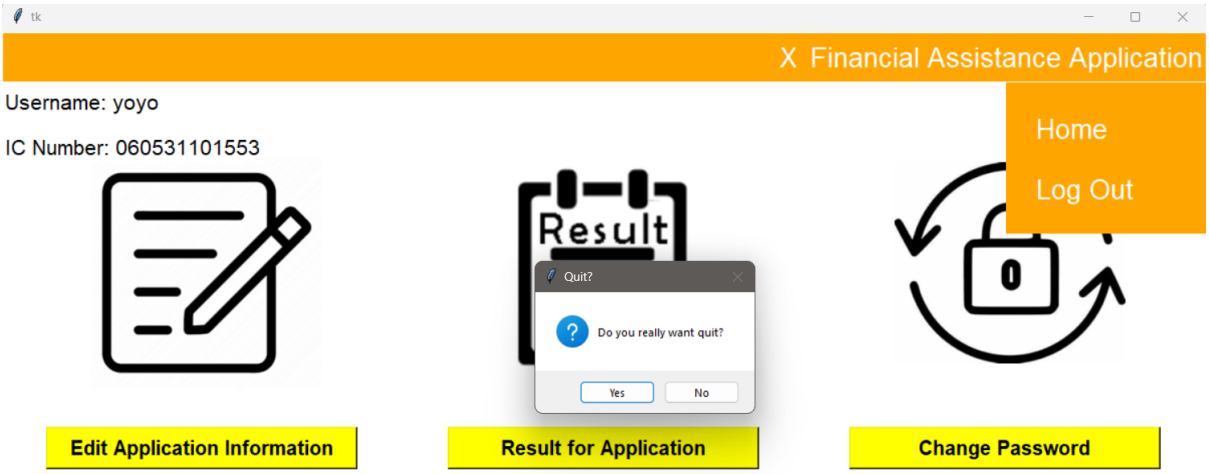
Page 5: After pressing Change Password button, it will turn to page 5. If the user keys in wrongly for confirm new password, it will show message box with the text ‘Your password didn’t get match!!’.



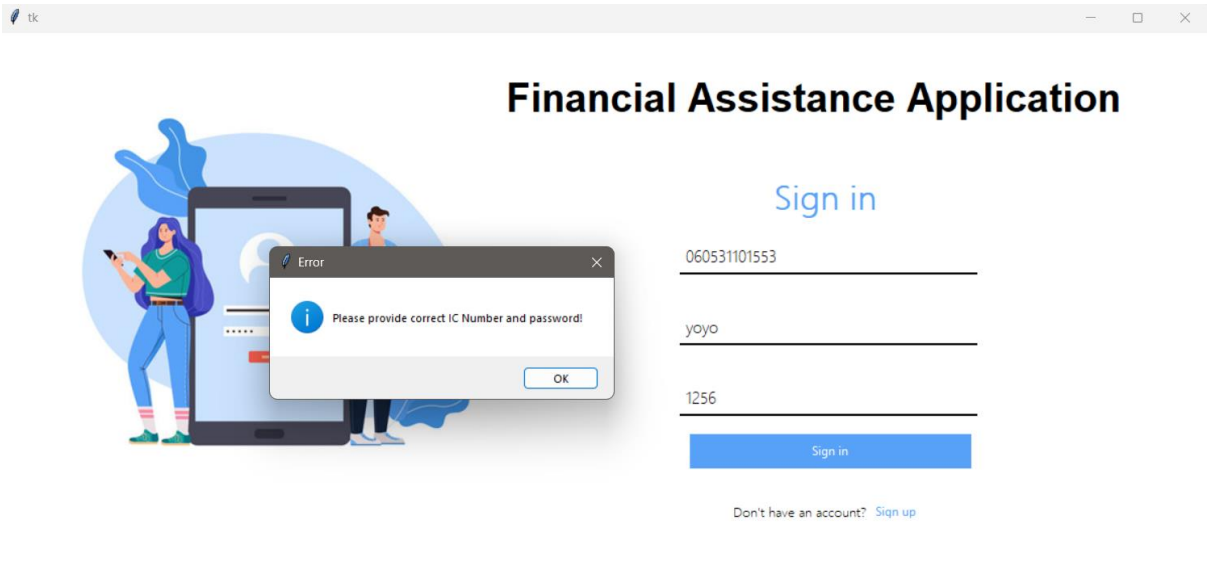
Page 5: After successfully changing the password, it will return to page 4.



Page 4: After pressing the ‘Log Out’, it show message box with the text ‘Do you really want quit?’. If pressing ‘yes’, the program will be terminated. If pressing ‘No’, it will stay at page 4.



Page 1: After pressing the ‘Home’ in page 4, it turn to page 1. If the user enter wrong IC number or wrong password, it will show message box with the text ‘Please provide correct IC Number and password!’. If uesr successfully log in, it will turn to page 4.



MARKING SCHEME

CLO	Description	PLO mapping	Percentage	Marks
CLO2	Use appropriate Python programming technique to solve problem.	PLO2: Cognitive Skills and Functional work skills with focus on Numeracy skills C3: Application	5%	10

LEVEL OF ACHIEVEMENT					
0	**1** **Inadequate**	**2** **Emerging**	**3** **Developing**	**4** **Good**	**5** **Excellent**

ELEMENTS	WEIGHTAGE	SCORE
Combination of appropriate controls and layout manager: • Input controls such as buttons, toggles, checkboxes etc. • Navigation controls such as pull-down menu. • Information components eg. message boxes etc. • Tkinter geometry manager (place/pack/grid manager).	**1**	
Task execution by each controls: • Each control is labelled using short and precise words representing the task. • The task for each controls is specified and written neatly. • The task for each control executed correctly and smoothly.	**1**	
TOTAL		

CLO	Description	PLO mapping	Percentage	Marks
CLO3	Construct and run program.	PLO3: Functional work skills with focus on Practical, and Digital skills P4: Mechanism	15%	30

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
Theory/ Knowledge	No theoretical knowledge is observed.	Very little knowledge provided or information is incorrect.	Some knowledge or information is provided but missing all major points.	Some knowledge or information is provided but still missing some major points.	Good knowledge is observed, missing some minor points.	Excellent knowledge is observed; provides all necessary background principles.	1	
Assembly	Fail to demonstrate the given task.	Partly demonstrate the given task with errors.	Partly demonstrate the given task with wrong output.	Partly demonstrate the given task correctly.	Fully demonstrate the given task with some wrong output.	Demonstrate the given task correctly and perfectly.	2	
Technique used / Effectiveness	Fail to demonstrate the given task.	Demonstrate inappropriate techniques.	Partly correct techniques demonstrated.	Demonstrated technique is correct but not effective or efficient.	Demonstrated technique is partly effective and efficient.	Demonstrated technique is effective and efficient.	2	
GUI	Not submitting GUI.	The GUI presented was taken from the other sources with no modifications. The GUI presented was not effective in debugging the output with a lot of errors and displayed for an inappropriate time.	The GUI presented was modified from the other sources with minimal modifications. Shows less effective debugging on the output with with several errors and displayed for less appropriate time.	The GUI presented was modified well from the other sources. Shows effective debugging on the output with no error and displayed for an appropriate time.	The GUI presented was modified very well from the other sources. Shows effective debugging on the output with no error and displayed for an appropriate time.	The GUI presented was originally developed. Shows effective debugging on the output with no error and displayed for an appropriate time.	1	

CLO	Description	PLO mapping	Percentage	Marks
CLO4	Work collaboratively to solve assigned task.	PLO4: Functional work skills with focus on Interpersonal skills A3: Valuing	5%	10

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
Foster Good Relationship	Show no good relationships and unable to work together effectively with other group members towards goal achievement.	No clear evidence of ability to foster good relationships and work together effectively with other group members towards goal achievement.	Able to foster relationship and work together with other group members towards goal achievement but with limited effect and require improvements.	Able to foster relationship and work together with other group members towards goal achievement with some effect(s) and require minor improvements.	Able to foster good relationship and work together with other group members towards goal achievement.	High ability to foster good relationship and work together effectively with other group members towards goal achievement.	1	
Alternate Roles	Show no ability to assume alternate roles as a group leader and group members.	No clear evidence of ability to assume alternate roles as a group leader and group members demonstrated in practice.	Attempt to demonstrate in practice the ability to alternate roles as a group leader and group members but with limited effect and require improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and group members with some effect(s) and require minor improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and a group member to achieve the same goal.	Show clear evidence to assume alternate roles as a group leader and a group member demonstrated in practice.	1	

CLO	Description	PLO mapping	Percentage	Marks
CLO5	Demonstrate innovative ideas in developing a graphical user interface.	PLO8: Entrepreneurial skills A3: Valuing	5%	10

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
Analyzing an existing situation and identifying areas for improvement	Not providing any analysis of situation and areas for improvement were not identified.	The analysis of the situation was very limited and areas for improvement were not. identified	The analysis of the situation was limited and areas for improvement were not identified.	The analysis of the situation was appropriate but the identification of areas for improvement was limited.	The situation was appropriately analyzed and the identification of areas for improvement was completed.	The analysis of the situation and the identification of areas for improvement was completed and increases over time.	1	
Creativity/ Innovative ideas	Not presenting any GUI.	GUI presented contains lack of significance ideas, no innovative values, lack of creativity and not user friendly.	GUI presented contains lack of significance ideas, no innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains lack of significance ideas, but still have innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains significance ideas, innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains a very significance ideas, high innovative values, creative enough (catchy apps name & attractive) and user friendly.	1	

END OF GROUP PROJECT'S INSTRUCTIONS