# TCS INFRAMIND-2019

## Artificial Intelligence

## "Team GCoEA"

## Requirement statement:

XFactory is a media company recently acquired another event management firm, total number of users after acquisition sum up to 30k. Our CEO's vision is to enhance user experience for better productivity which will directly impact company's revenue and profit. As part of this initiative, IT manager decided to build a platform which increase productivity of an employee without affecting his/her interest. How to increase the user experience index will be the core requirement for XFactory media company and based on multiple discussion and surveys , IT managers decided to introduce a BOT (desktop version / Platform ) and raised a request to Ux ( User Experience ) enhancement team to develop BOT to perform following actions as part of user experience initiative,

## BOT Requirement:

- User should be notified about his/her meeting schedules on timely basis
- Share live updates about user's favorite sport or events
- User shall be informed about current happens in the share market
- Keep user up to date about corporate notifications
- User should have fun learning with BOT about health tips
- Share Facebook updates
- Share /notice twitter trends

## Code:

| Third Party Packages | Purpose |
|---|---|
| tkinter | Making GUI |
| time | Generating delay |
| nsetools | Generating Share Market Updates |
| sports | Generating favourite sports updates |
| win10toast | Generating Notifications |

```
import tkinter
from tkinter import ttk

# used Tkinter package of python
window = tkinter.Tk()

# GUI title
window.title("INFRAMIND")

# geometry of the window that will open
# pixel size of window "X x Y"
window.geometry("300x250")

# Form data - Firt name field
labelOne = ttk.Label(window, text="First Name")
labelOne.grid(row=0, column=0)

# Form data - last name field
labelTwo = ttk.Label(window, text="Last Name")
labelTwo.grid(row=1, column=0)

# Form data - Favorite sports field
labelThree = ttk.Label(window, text="FavSport")
labelThree.grid(row=2, column=0)

# Form data - Required Stock update company
labelFour = ttk.Label(window, text="Stock Update")
labelFour.grid(row=3, column=0)

# Form data - Enter gender
labelFive = ttk.Label(window, text="Gender")
labelFive.grid(row=4, column=0)

# Form data - Enter Age
labelSix = ttk.Label(window, text="Age")
labelSix.grid(row=5, column=0)

# Form data - Enter Notification time
labelSeven = ttk.Label(window, text="Notification
time")
labelSeven.grid(row=6, column=0)

# Form data - Enter time interval between successive
updates
```

```
labelEight = ttk.Label(window, text="Interval time")
labelEight.grid(row=7, column=0)

# getting the user data
userFName = tkinter.StringVar()
userLName = tkinter.StringVar()
sport = tkinter.StringVar()
company = tkinter.StringVar()
gender = tkinter.StringVar()
age = tkinter.StringVar()
Notf_t = tkinter.StringVar()
Int_t = tkinter.StringVar()

# storing the entered data of the user
userEntry = ttk.Entry(window, width=26,
textvariable=userFName)
userEntry.grid(row=0, column=1)

userEntry = ttk.Entry(window, width=26,
textvariable=userLName)
userEntry.grid(row=1, column=1)

# generate a drop-down list for available sports
set1 = tkinter.OptionMenu(window, sport, "baseball",
"basketball", "cricket", "football", "handball",
"hockey",
                          "rugby-league", "rugby-
union", "soccer", "tennis", "volleyball")
set1.grid(row=2, column=1)

# generate a drop-down list for avaible companies
(more number of companies can be added)
set2 = tkinter.OptionMenu(window, company,
"ADANIPORTS", "HINDALCO", "ONGC", "SBIN")
set2.grid(row=3, column=1)

userEntry = ttk.Entry(window, width=26,
textvariable=gender)
userEntry.grid(row=4, column=1)

userEntry = ttk.Entry(window, width=26,
textvariable=age)
userEntry.grid(row=5, column=1)
```

```
userEntry = ttk.Entry(window, width=26,
textvariable=Notf_t)
userEntry.grid(row=6, column=1)

userEntry = ttk.Entry(window, width=26,
textvariable=Int_t)
userEntry.grid(row=7, column=1)

labela = ttk.Label(window, text="secs")
labela.grid(row=6, column=2)

labelb = ttk.Label(window, text="secs")
labelb.grid(row=7, column=2)


# action function
def action():
    print(userFName.get())
    print(userLName.get())
    print(sport.get())
    print(gender.get())
    print(age.get())
    print(Notf_t.get())
    print(Int_t.get())
    window.destroy()


# Submit button for the form . Calls action method
for further process
btn = ttk.Button(window, text="Submit",
command=action)
btn.grid(row=8, column=0)

# open the window
window.mainloop()

# infinite loop
while 1:
    # for generation of delay
    import time

    # displaying the stock market data
    from nsetools import Nse
```

```
    # gettinfg the sports data
    import sports

    from win10toast import ToastNotifier

    nse = Nse()

    # stock name and buy price pair, but not used in
my method below. I would like a method that uses it
though.
    stocks = {
        "ADANIPORTS": 284.95,
        "HINDALCO": 152.30,
        "ONGC": 279.70,
        "SBIN": 259.70
    }

    # stock name and quantity of stocks
    qty = {
        "ADANIPORTS": 20,
        "HINDALCO": 20,
        "ONGC": 20,
        "SBIN": 20
    }
    ### Adaniports
    ### Hindalco
    ### ONGC
    ### SBIN
    # Fetches a dictionary with data such as best
buy/sell prices, 'closePrice', etc.
    stock = nse.get_quote(str(company.get()))

    # Accessing the 'closePrice' from the above
dictionary.
    stock1Close = stock['closePrice']

    n = str(stock)

    x = sports.all_matches()
    c = x[sport.get()]
    m = str(c)

    notifier = ToastNotifier()
```

```
    notifier.show_toast(sport.get(), m,
duration=int(Notf_t.get()))

    notifier.show_toast(company.get(), n,
duration=int(Notf_t.get()))

    time.sleep(int(Int_t.get()))
```

**YouTube Video Link:**

https://youtu.be/xt6n6Yap2jE

**Back-up Drive link:**

https://drive.google.com/drive/folders/1mY39OkXBaoTDkNs0Io2VQ0VcKxGEqlTe

**(If the link does not work, kindly paste the link in the address bar)**

**Work in progress:**

- **Conversion of python script to executable file using pyinstaller.**
- **API generation for Social medias like Twitter, Facebook, Google Calendar and Gmail alerts.**
- **Generation of Health Notifications.**
- **Use of Database to enable personalised authorisation and communication within the login community.**