# 

# 

# 

# PATAN COLLEGE FOR PROFESSIONAL STUDIES

# Assessment 2- Group Assignment

# Python Project

# 

# Shahu: The Ultimate Shared Power

# PROJECT REPORT

# Submitted by Submitted to

# Kshitij Bajagain (1723333) UOB

# Dinesh Timalsina (1815429) Luton, Bedfordshire

# Ram Ghimire (1816707)

# Anupam Dhakal (1815430)

**Abstract:**

**Shared power** is an information system which enables the function of sharing the expensive and specialist tools rather than buying them in order to help tradesmen to save money. Before developing this system, proper and vague study was done to figure out the difficulties of tradesmen regarding inability to buy expensive tools. The best solution to solve this problem of tradesmen was to develop a shared power where the tradesmen can add tools, search tools, hire tools, and return tools rather than buying expensive new tools. We developed a software named Shahu: The Ultimate Shared Power which will solve problem of tradesmen regarding inability to buy expensive tools.

**Acknowledgement:**

We would like to show gratitude towards all the people who helped us during development of this project. At first, we would like to thank our unit leader, project supervisor BishwoKiran Bhattarai, HOD Dr. Swati Shah, and Dean Academics Er. Ajaya Sharma for supervising and guiding us during whole software development phase.

In addition, we would also like to thank our colleagues and all other people who helped and guided us during the research as well developing phase of this software. And also, we would like thank them for their valuable time for testing our software.

**Dedication:**

We are feeling proud to dedicate this software to those tradesmen who were deprived of buying expensive tools for their profession**.**

**Table of Contents:**

* Abstract
* Acknowledgement
* Dedication

1. Introduction
   1. Aim of project
   2. Proposed System
2. System Design
   1. Graphical User Interface(GUI) design
   2. Text File Handling Approach
3. System Architecture
   1. UML Diagrams
      1. Sea-Level Use Case Diagram
      2. Scenarios(Fish-Level Diagram)
      3. Clam Level Use Case Diagram
      4. Class Diagram(OOPS – oriented approach)
      5. Activity Diagram
4. Implementation Of program
   1. Screenshots of Software
   2. Software used for development
5. Group Organization and Role division
6. Software Development
   1. Code
   2. Testing
7. Conclusion

**1. Introduction:**

**Shared power** is an information system which enables the function of sharing the expensive and specialist tools rather than buying them in order to help tradesmen to save money. There are some system terms and conditions that the user needs to follow in order to use shared power system. User need to be registered with correct details by filling our registration form. After getting registered into our system registered user can upload tools, search tools, hire tools, return tools and can preview/print invoice at last day of every month on the basis of their hired tool info. While hiring tools, registered user can only hire tools for maximum of 3 days, late return of tools will be penalized by doubling the day rate of tool every day. While generating the monthly invoice, extra £5 is added to the total amount of specific hired tool. In our system, there is a default account for Insurance Agent which will analyze the tool info as well as the registered user info for ensuring the quality of tool hired and uploaded.

* 1. **Aim/Objective of project**

The main aim of this Shahu: The Ultimate Shared Power project is to solve the problem regarding inability of tradesmen to buy expensive tools for their professional use. Shahu: The Ultimate Shared Power’s objective is to make easy and convenient way of transition (hiring) of tools between tradesmen.

* 1. **Proposed System**

The system that we proposed is Shahu: The Ultimate Shared Power which is an information system developed by using Tkinter library of Python language. On the basis of assignment given by University of Bedfordshire, we have designed and developed this system to solve that specific mentioned problem of tradesmen. This system provides the easy and convenient way of hiring tools for tradesmen solving the problem of buying expensive tools.

1. **System Design**

Designing a system is always a critical phase during software development. System design consists of proper detailed planning of developing, maintaining, altering and finishing the system. Software Development Life Cycle (SDLC) is the main approach which helps to develop and design high quality system. Quality system always implement SDLC approach, we have also followed the SDLC guidelines and formulated the idea and plans during software development and designing phase. In this shared power system we have implemented and used USECASES, CLASS DIAGRAM, and ACTIVITY DIAGRAM. We have used following methodology and process.

1. **Brainstorms and Planning:** Brainstorms are always important during development of software. We did brainstorms and figure out the perfectplanningfor our group project.
2. **Design and Analysis:** After brainstorms and planning we were all set for designing the software. We designed our system using GIMP and implemented those designs in basic GUI framework of Python i.e. Tkinter.
3. **Implementation:** We implemented all those design, brainstorms and planning during coding process of software development cycle.
4. **Testing:** After all process of planning, designing and coding was over we tested our software in order to identify possible errors and bugs.
5. **Feedback and Evaluation** : Evaluation and feedback is another important process because after completion project, feedbacks and review from other peoples has motivated us to do more and better in upcoming project.
   1. **Graphical User Interface (GUI) design**

System designing is the crucial during the software development phase. GUI is the main feature of software which attracts the user to implement the software’s functions without being bored. We scheduled sufficient time to develop graphical user interface making ensure that user never gets bored using our software for tool transition. We used **GIMP** for designing the image used in software and **visual paradigm** as well as **draw.io** for designing UML diagrams.

**2.2. Text File Handling Approach**

Making software requires equal involvement of every phase. File handling is another main phase in python software development which helps to store all the data in text file format in order to run the perfect flow of program. We used normal file handling to write, read, append, delete, change and manipulate the data entered by users in this system.

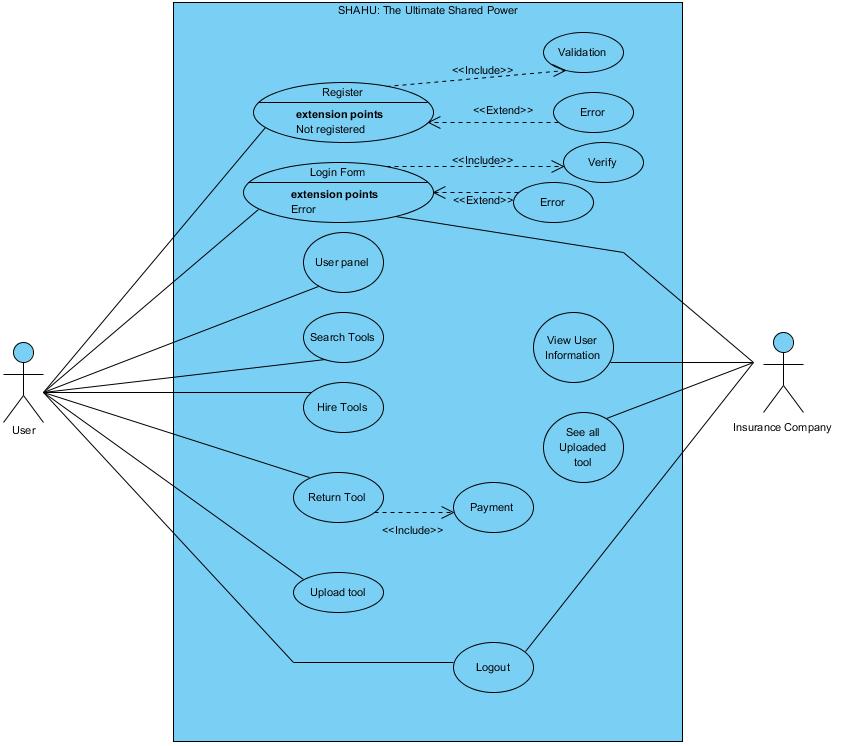
1. **System Architecture**

System Architecture is the most important during development phase of software. We have implemented following UML diagrams and SDLC approach during development phase.

* 1. **UML Diagrams**

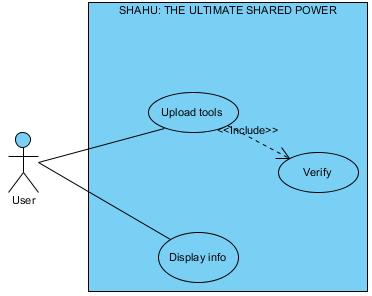
We used Unified Modeling Language for visualizing, constructing, documenting and specifying the components of this software. Every diagram that we used during development of software is shown below:

* + 1. **Sea-Level Use Case Diagram**



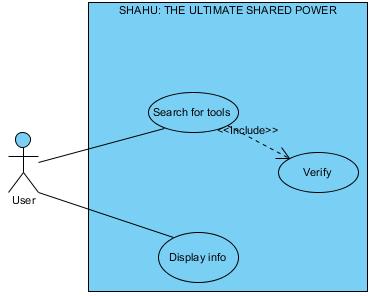
This diagram is Sea Level Use Case Diagram. Sea level mentioned above consists of all use cases which we used for developing this system.

* + 1. **Scenarios(Fish-Level Diagram)**
* **Scenario 1:**

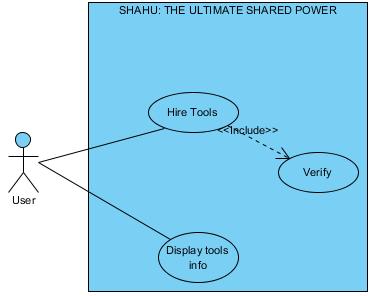


**Registered user can upload tools in our software database so that other registered user can hire those specific tools on the basis of uploaded tool info.**

* **Scenario 2:**

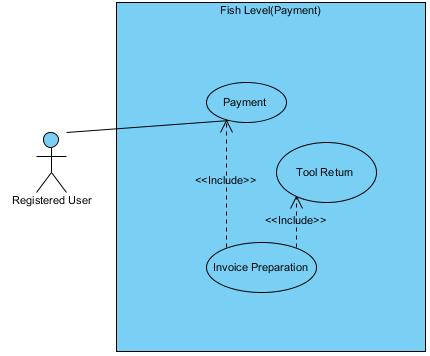


**Registered user can search uploaded tools in our system which is added by other registered user in our system database.  
Scenario 3:**

****

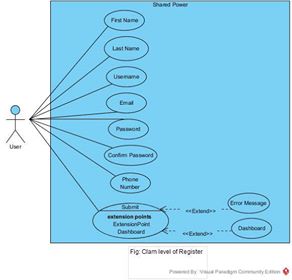
**Registered user can hire tools from our software. They can hire only uploaded tools by other registered user in our system.**

**Scenario 4:**

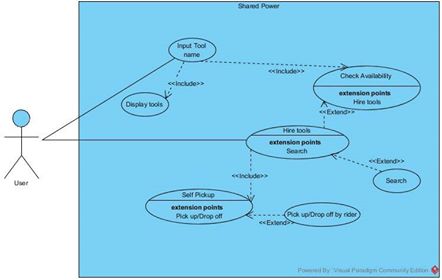


**Registered user can generate invoice with insurance tax included instantly from our system which will be delivered while return of tools.**

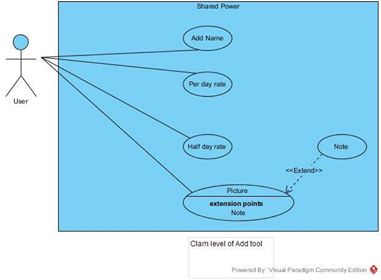
* + 1. **Clam Level Use Case Diagram**

****

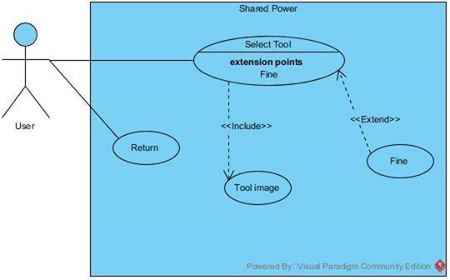
**Clam level diagram of Register**

****

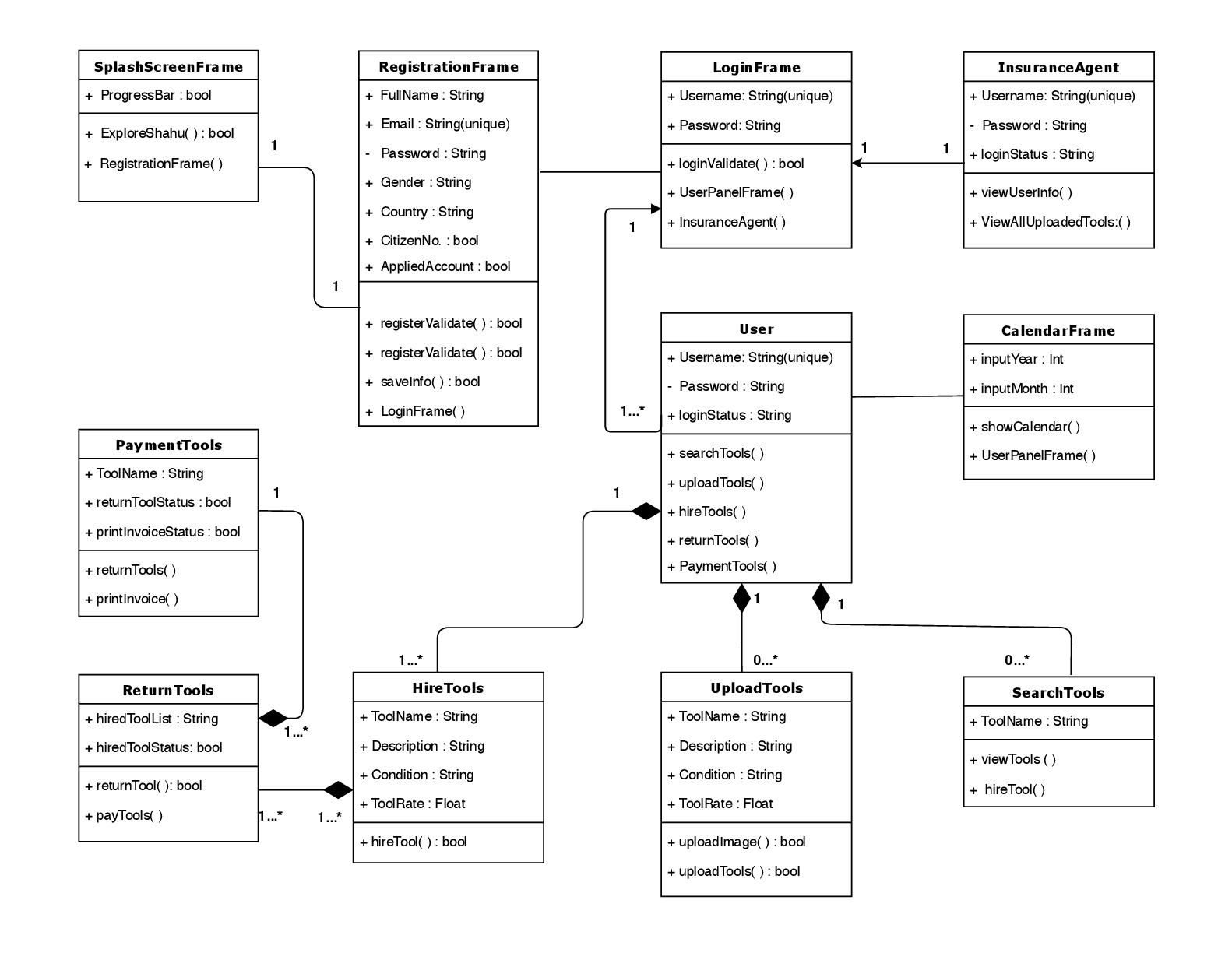
**Clam level diagram of search and hire tool**

****

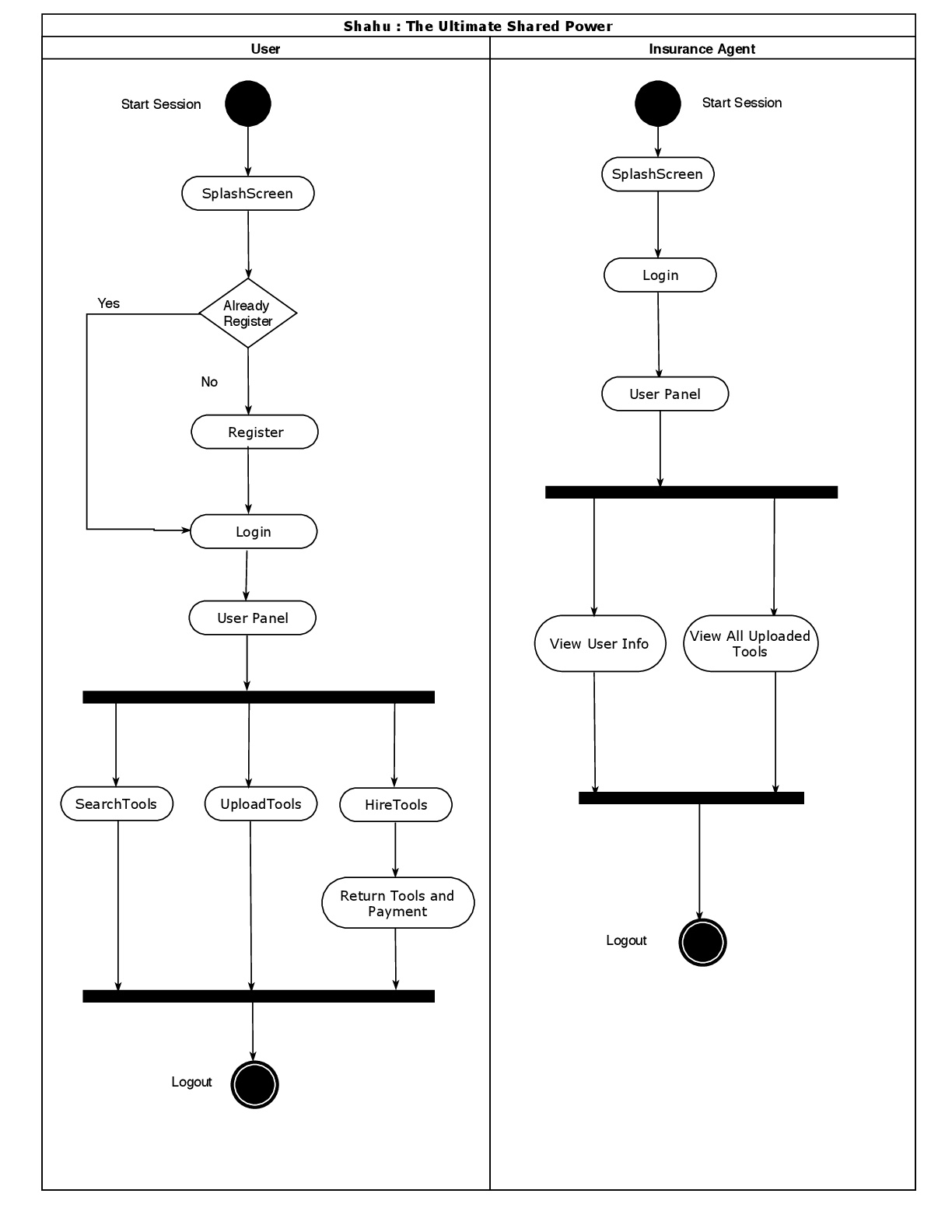
**Clam level diagram of Upload Tools**

**Clam level diagram for return tool.**

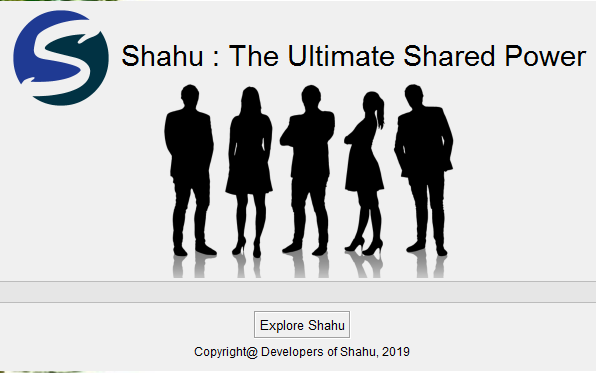
* + 1. **Class Diagrams(OOPS – oriented approach)**



* + 1. **Activity Diagrams**

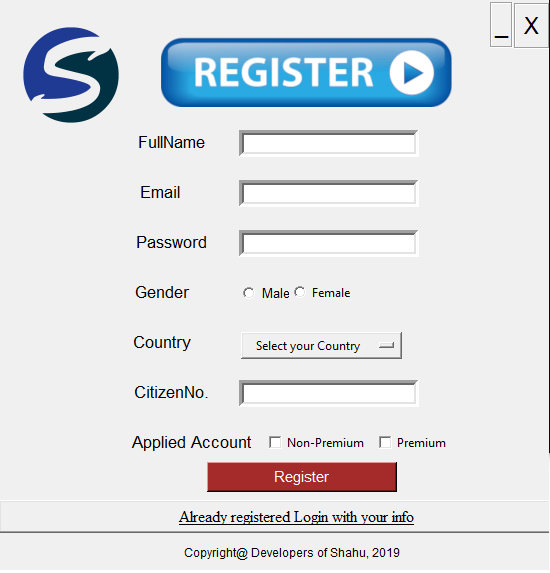
****

1. **Implementation Of program**
   1. **Screenshots of Software**

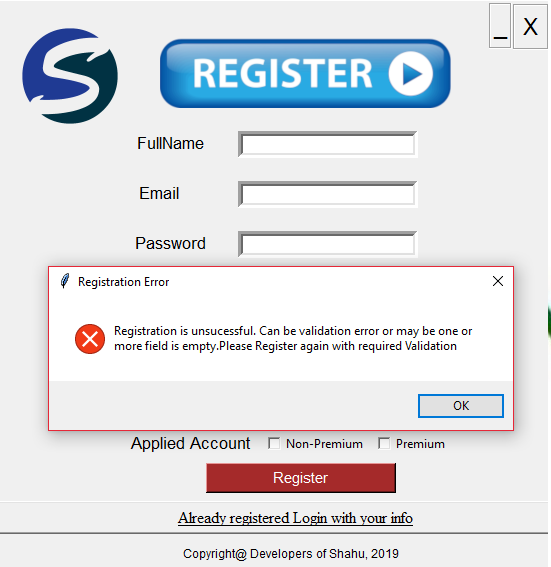
****

**This is Splash Screen of software which is used for initializing software.**

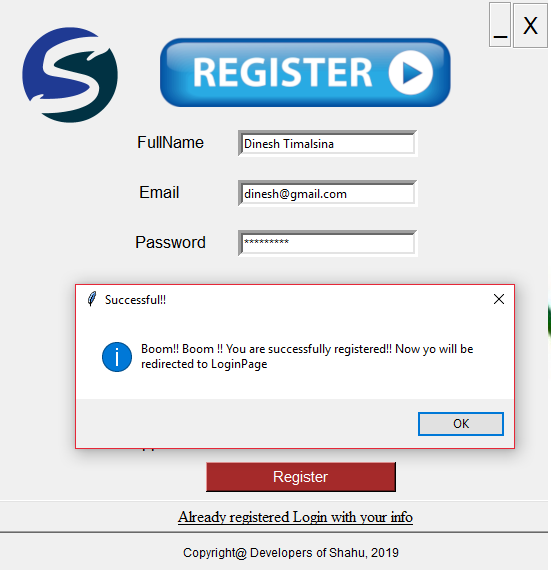
**REGISTER FORM**



**Users can enter their information in the register page and can be the member of Shahu.**

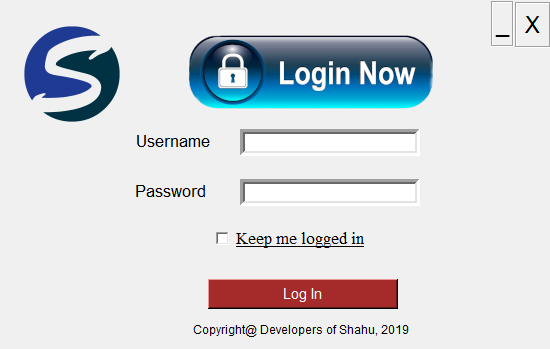
****

**Validation check for registration page. Empty field are strictly denied in our system.**

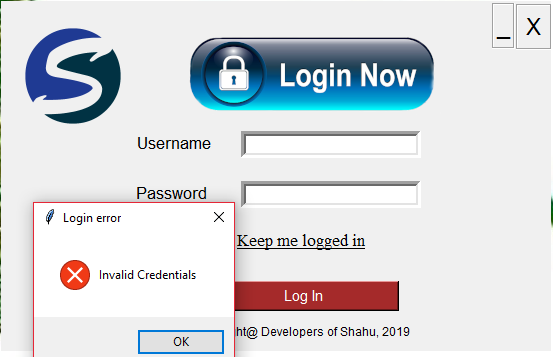
****

**Successful registration can be done by giving correct information and will redirected to login page.**

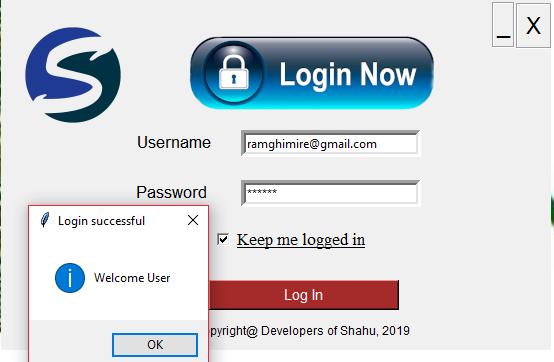
**LOGIN PAGE**

****

**This is login page where user can enter their credentials in order to start their session in our software.**

****

**Invalid credentials or empty credentials error check.**

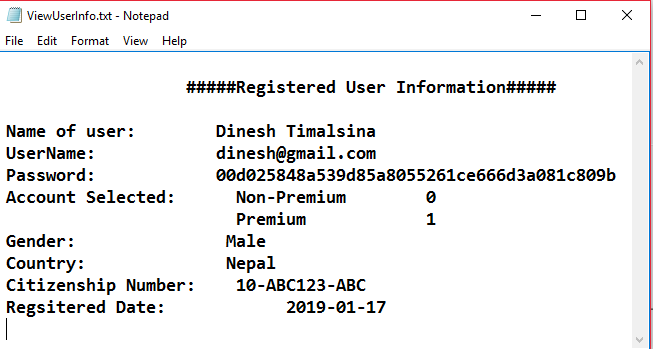
****

**User with valid credentials can log in into our software successfully.**

**USER PANEL**

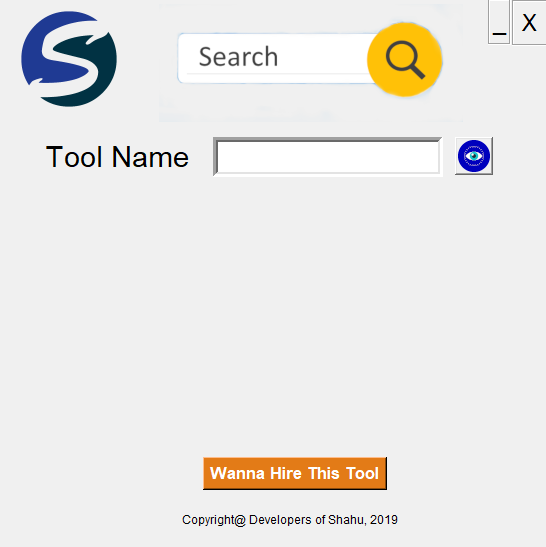
****

**This is user panel where a valid user can search tools, upload tools, hire tools and can preview/print invoice by using our system.**

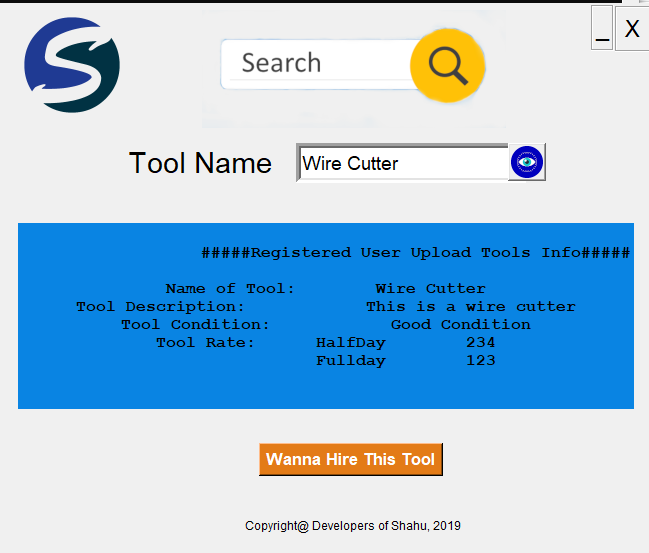
****

**When user clicks the view user info button then he will get the function of viewing his registered detail.**

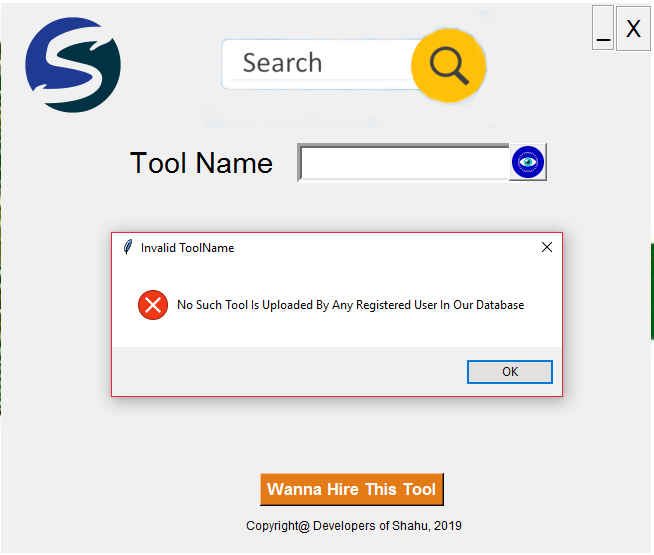
**SEARCH TOOLS FUNCTION**

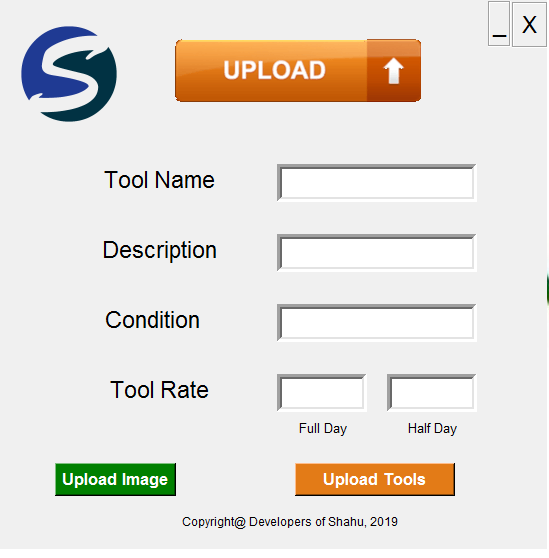


**Valid user can search tools uploaded by other valid users and hire tool.**

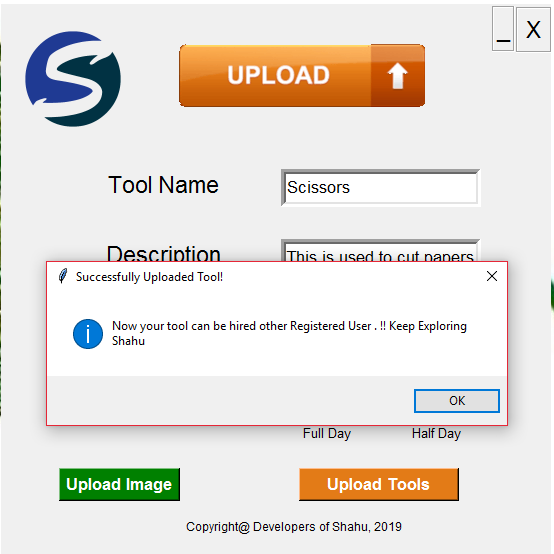


**Successfully searched the tool and ready to hire searched tool.**

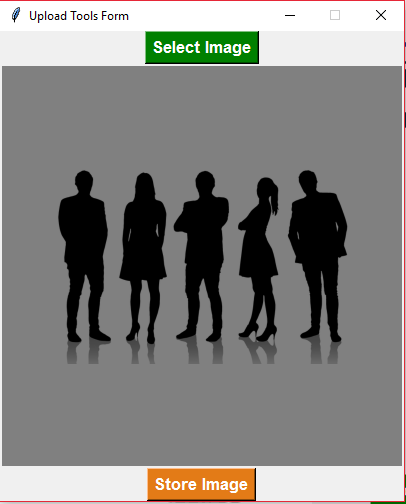
**Error occurred when empty field was empty.  
**



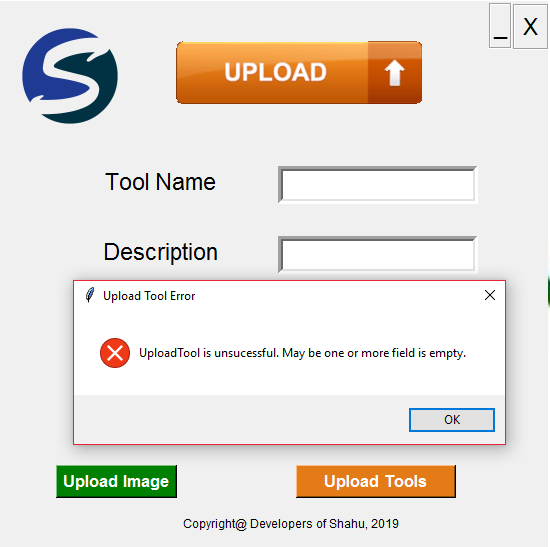
**This is upload tools functions where registered user can upload tools by entering tools details.**

****

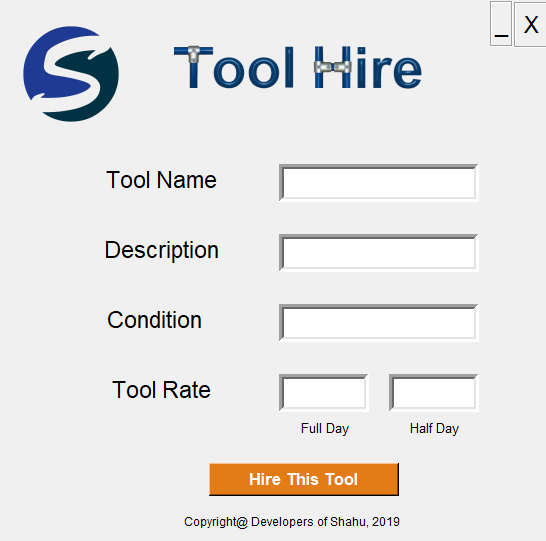
**Successful upload of tools by giving valid details.**

****

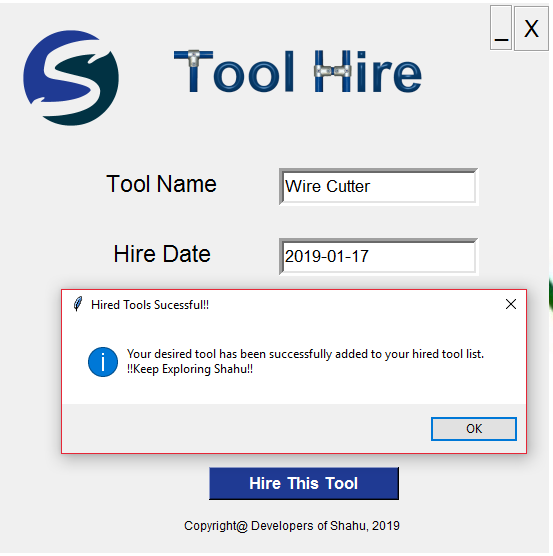
**Upload Image function which will store the image in our system database .**

****

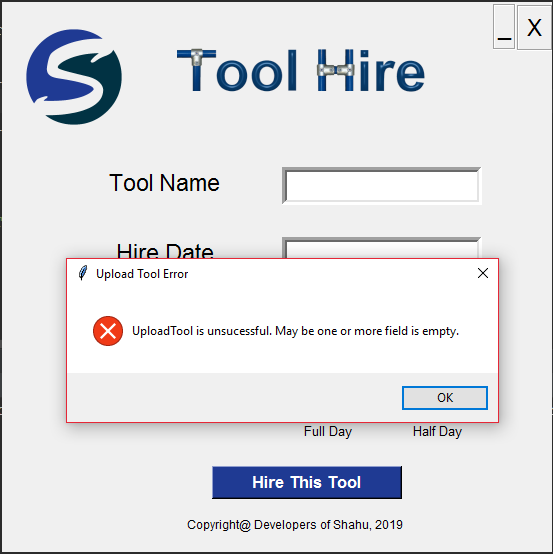
**Error occurred when upload tools field were left empty.**



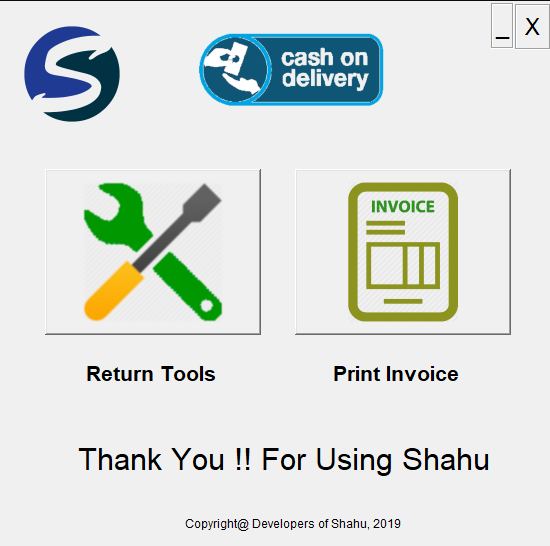
**Registered user can hire the tool uploaded by other user in our database.**

****

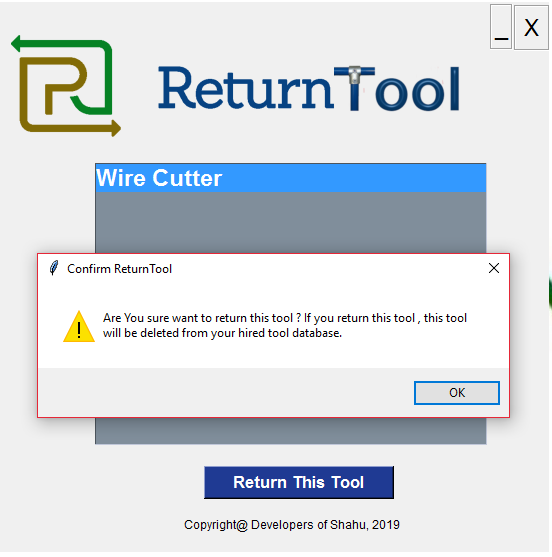
**Successfully hired the tool by inserting valid details .**

****

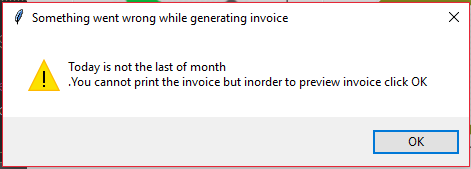
**Validation check , empty fields or invalid details cause error during hiring tool.**



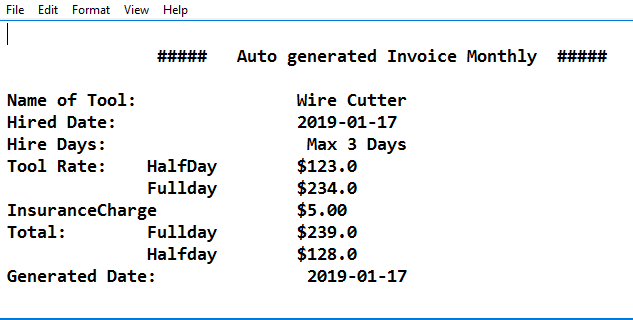
**This is return tools where registered user can return tools and use invoice generation function.**

****

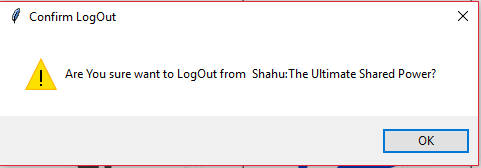
**Registered User can return hired tools before printing invoice**

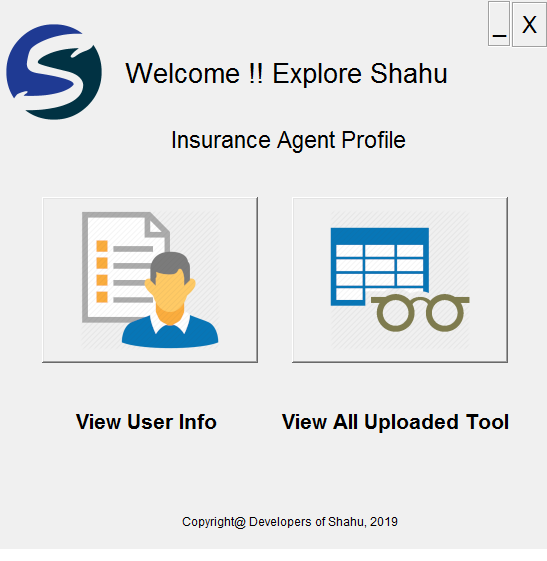
****

**Messagebox Invoice generation**

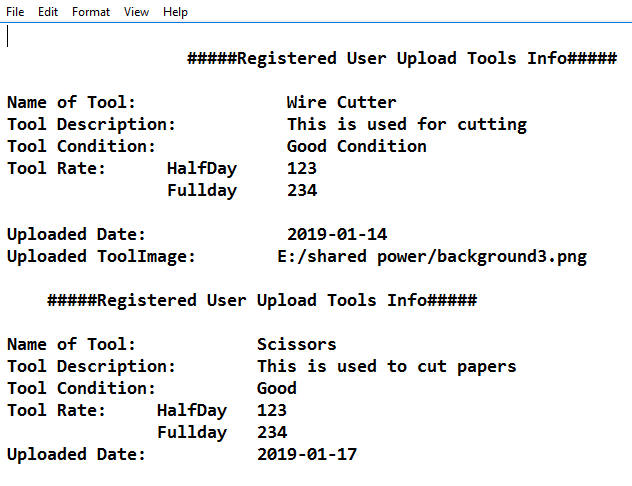
****

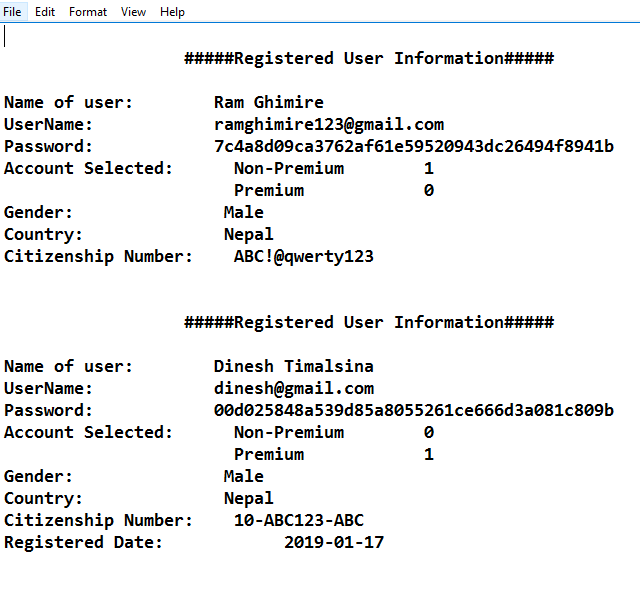
**Invoice generation**

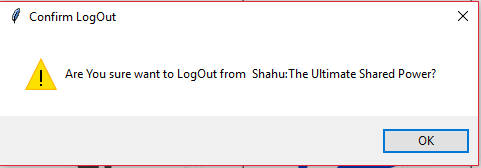
**Log out validation check Message box occurs when somebody log outs.**



**Insurance Agent profile where he has two functions.**

**View All Uploaded tool info**

**View All Registered user info.**

**Insurance agent Log Out from our system**

* 1. **Software used for development:**

For the development of the whole software we used software like: GIMP 2.10.8 and Photoshop CC for editing of image and pictures. These software’s were used for development of high quality GUI images .And we used python 3.7.1 and Pycharm professional version on university education license for coding.

1. **Group Organization and Role division**

We planned to divide the role during the initial phase of development of software. There were some confusion in initial phase of development regarding the role but everything were solved .Following table shows the role division of our project. This role division was strictly followed while development phase.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Name** | **Performed Roles** | **Student ID** |
| **1.** | Kshitij Bajagain | Developer | 1723333 |
| **2.** | Anupam Dhakal | Designer | 1815430 |
| **3.** | Dinesh Timalsina | Tester | 1815429 |
| **4.** | Ram Prasad Ghimire | Analyst | 1816707 |

**Group Division**

Following mentioned table below is the work done by individual during the software development.But during presentation, to manage time, diagrams were explained by Kshitij .He has mentioned clearly in presentation that activity diagram is done by Ram and Dinesh and Use case Diagrams (sea, fish level) was done by Anupam and Class diagram was done by Kshitij.

|  |  |  |
| --- | --- | --- |
| **S. N** | **Name** | **Work Division** |
| **1.** | Kshitij Bajagain | User Panel, Search Tools, Hire Tools, Upload Tools , Class Diagram |
| **2.** | Anupam Dhakal | Login , Register, UML Diagrams(Sea level, Fish Level), Slides |
| **3.** | Dinesh Timalsina | Return Tools, Payment , Invoice Generation, Clam level Diagram |
| **4.** | Ram Prasad Ghimire | Insurance agent profile design, Activity diagram, Analysis of software |

1. **Software Development** 
   1. **Testing**

A>Testing on Registration

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SN | | Action | | Input value | Expected Result | Actual Result | Status |
| 1 | | Registration | | All field such as first name, last name, Email, cabin no, username and password are inserted | User should be able to register successfully to the system | User registered successfully | pass |
| 2 | | Registration | | All or few field from the registration form are left blank | Error is seen | Invalid particular blank portion | pass |
| 3 | | Fill full name | | Insert full name with string | Valid full name | Valid full name because full name contains only string | pass |
| 4 | | Fill Email | | Email field blank  Email with no @ and .com  Valid email like test@gmail.com | Invalid email  Invalid email  Email successfully inserted | Message displayed invalid email address  Message displayed invalid email address.  Email is accepted. | pass |
| 5 | Password field | | Password field blank  Special symbols and characters in password field | | Invalid password  Password is accepted | Invalid message is seen  Password is accepted | pass |

B>Testing on Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Action | Input value | Expected result | Actual result | Status |
| 1 | Login | If username and password both field are inserted | Login successful | Login was successfully done as excepted. | Pass |
| 2 | Login | If one or both field are left blank | Error is seen | Invalid username and password message is displayed. | Pass |
| 3 | Fill username | Username filled blank  Test | Invalid username  Username accepted | Same as expected result | Pass |
| 4 | Fill password | Password field blank  Special symbols and characters in password field | Invalid password  Password is accepted | Invalid message is seen  Password is accepted | Pass |

**C> Search Tools, Hire Tools, Upload Tools**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Action | Input value | Expected result | Actual result | Status |
| 1 | Search Tools | If correct uploaded tool was searched | Search Tool successful | Search tool was successfully done as excepted. | Pass |
| 2 | Search Tools | If one or both field are left blank | Error is seen | Invalid tool name. | Pass |
| 3 | Upload Tools | If valid details are filled in boxes | Upload tools was successful. | Same as expected result | Pass |
| 4 | Upload Tools | If one or more field was left empty | Upload tools error was occurred | Same as expected | Pass |
| 5 | Hire Tools | If valid details are filled in boxes | Hire tools was successful. | Same as expected result | Pass |
| 6 | Hire Tools | If one or more field was left empty | Hire tools error was occurred | Same as expected | Pass |

**Conclusion:**

After completion of project, we are highly motivated to do better in upcoming project. We think that we have solved the problem given by university regarding the tradesmen inability of buying expensive tools. We have developed this software in python language and using basic Tkinter library of python. For image designing and manipulation we used GIMP and we used draw.io and visual paradigm for developing UML diagrams. We hope that we will continue our passion in developing software’s and be more competitive and make more advanced software in upcoming future. We will like to thank everybody for supporting us during development phase. At last , we are proud of ourselves and we hope we have solved the proposed problem by University of Bedfordshire.

**Codes:**

**"""**

**This Shahu: Ultimate Shared Power software is made in tkinter library which is basic native GUI framework of Python.**

**Developers::1>Kshitij Bajagain -Main developer(coder) (kshitij.bajagain123@gmail.com, 2019)**

**2>Anupam Dhakal -Documentation,2nd-coder (anupamdhakal20@gmail.com, 2019)**

**3>Dinesh Timalsina -Tester,Idea generator (timalsinadinesh2@gmail.com, 2019)**

**4>Ram Ghimire -Analyst (ramghimire733@gmail.com, 2019)**

**Python Interpreter = Python 3.7.1**

**IDLE Used = JetBrains Pycharm Professional(University License agreement)**

**Designing software used = GIMP 2.10.8 and Photoshop 2018 CC**

**Copyright = Illegal copy and distribution of this software is strictly prohibited .**

**Developers can only change and make copy of this Shahu:Ultimate Shared Power Software**

**"""**

**## These are all library that we used during development of this software##**

**from tkinter import \***

**import tkinter as tk**

**import tkinter.messagebox as tm**

**from tkinter import Frame**

**import tkinter.messagebox**

**from tkinter import ttk**

**import time**

**import json**

**from datetime import date**

**import os**

**import calendar**

**import hashlib**

**from PIL import Image, ImageTk**

**from tkinter import Tk, filedialog**

**'''**

**This is SplashScreen Class which will initialize the software and redirects you to Register Page**

**'''**

**class SplashScreenFrame(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master, padx=2, pady=2)**

**self.button = Button(master,text="Explore Shahu", font=("bold", 10), relief="groove", activebackground="green",command=self.command1).place(x=255, y=310)**

**self.progress\_bar = ttk.Progressbar(master, orient="horizontal",**

**length=600, mode="determinate")**

**self.progress\_bar.place(x=0,y=280)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=90, y=340)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**title = Label(root, text="Shahu : The Ultimate Shared Power", width=30, font=("bold", 22))**

**title.place(x=97, y=35)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=10, y=5)**

**self.img1 = PhotoImage(file="Images\Background3.png")**

**self.lab1 = Label(master, image=self.img1)**

**self.lab1.place(x=145, y=75)**

**def open\_terms(self):**

**os.startfile("Terms.txt")**

**def command1(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**root.state('withdrawn')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = RegistrationFrame(self.newWindow)**

**self.newWindow.geometry('550x570+450+110')**

**self.newWindow.title("Shared Power Registeration Form")**

**def run(self):**

**self.progress\_bar.start()**

**#self.progress\_bar['maximum'] = 100**

**'''**

**This is LoginFrame Class which will login user into their respective UserPanels .Registered User Can enter their**

**credentials inorder to start their session in our software.**

**'''**

**class LoginFrame():**

**def \_\_init\_\_(self, master):**

**global entry\_username, login\_password**

**self.master = master**

**self.frame = tk.Frame(master)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17), relief="groove", activebackground="blue",**

**command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_destroy = Button(master, text="X", width=2, font=("bold", 17), relief="groove", activebackground="red",**

**command=endProgram)**

**self.button\_destroy.place(x=515, y=3)**

**self.label\_username = Label(master, text="Username", width=20, font=("bold", 13))**

**self.label\_username.place(x=80, y=130)**

**self.entry\_username = Entry(master, bd=5)**

**self.entry\_username.place(x=240, y=130, width=180)**

**self.label\_password = Label(master, text=" Password", width=20, font=("bold",13))**

**self.label\_password.place(x=68,y=180)**

**self.login\_password = Entry(master, bd=5,show="\*")**

**self.login\_password.place(x=240, y=180, width=180)**

**var2 = IntVar()**

**Checkbutton(master, text="Keep me logged in", variable=var2, font=('Times', 13,'underline'),cursor="hand1").place(x=190, y=225 , width=200)**

**Button(master, text='Log In', font=("bold", 11), width=20, bg='brown', fg='white', command=self.login\_info).place(**

**x=208, y=280)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=20, y=23)**

**self.img1 = PhotoImage(file="Images\login2.png")**

**self.lab1 = Label(master, image=self.img1)**

**self.lab1.place(x=180,y=30)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=88, y=320)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def open\_terms(self):**

**os.startfile("Terms.txt")**

**'''**

**This login\_info method will save the login info of user and as well as this will check the**

**credentials inorder to start their user session in our software.**

**'''**

**def login\_info(self):**

**self.passw1=self.login\_password.get()**

**self.Username\_Login = self. entry\_username.get()**

**self.Password\_Login = hashlib.sha1(self.login\_password.get().encode()).hexdigest()**

**file = open("Text File Handling\database3.txt", "r")**

**self.login1=file.read()**

**if self.Username\_Login and self.Password\_Login in self.login1:**

**tm.showinfo("Login successful", "Welcome User")**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+15')**

**self.newWindow.title("Shared Power Login Form")**

**elif self.Username\_Login == "Insurance!" and self.passw1 =="INS!@#":**

**self.master.withdraw()**

**tm.showinfo("Login successful", "Welcome to Insurance Company Profile ")**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = InsuranceCompany(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Insurance Company Form")**

**else:**

**tm.showerror("Login error", "Invalid Credentials")**

**def minimizeProgram(self):**

**#self.master.wm\_state('iconic')**

**self.master.withdraw()**

**def \_login\_btn\_clicked(self):**

**# print("Clicked")**

**username = self.entry\_username.get()**

**password = self.login\_password.get()**

**# print(username, password)**

**if username == "Shahu" and password == "admin":**

**tm.showinfo("Login info", "Welcome User")**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+15')**

**self.newWindow.title("Shared Power Login Form")**

**else:**

**tm.showerror("Login error", "Invalid Credentials")**

**'''**

**This is UserPanel Class. After getting loged in with correct credentials the user will be redirected to this**

**panel.This user Panel gives the user the function like searching the tool , uploading the tool, hiring the tool**

**, returning the tool, and generating the invoice by clicking on respective functions.**

**'''**

**class UserPanelFrame(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master)**

**self.title = Label(master, text=" Current Time", font=("bold", 11))**

**self.title.place(x=605, y=80)**

**self.q = Canvas(master)**

**self.q.place(x=600, y=97)**

**self.localtime = time.asctime(time.localtime(time.time()))**

**self.q.create\_text(14, 28, text=self.localtime, font=('arial', 12, 'bold'))**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17),relief="groove",activebackground="blue",command=self.minimizeProgram)**

**self.button\_minimize.place(x=661, y=2)**

**self.button\_destroy = Button(master, text="X", width=2, font=("bold", 17) , relief="groove" , activebackground="red",command=endProgram)**

**self.button\_destroy.place(x=685, y=3)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9),cursor="hand2")**

**self.label\_6.place(x=196, y=682)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.title = Label(master, text="Welcome !! Explore Shahu ", width=30, font=("bold", 22))**

**self.title.place(x=110, y=35)**

**self.tool = PhotoImage(file="Images\Tool1.png")**

**self.tool1 = Label(master, image=self.tool)**

**self.tool1.place(x=125, y=90)**

**self.tool1 = PhotoImage(file="Images\Tool1.png")**

**self.tool12 = Label(master, image=self.tool1)**

**self.tool12.place(x=520, y=90)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=10, y=5)**

**#self.img3 = PhotoImage(file="Sp1234.png")**

**#self.lab1 = Label(master, image=self.img3)**

**#self.lab1.place(x=590, y=620)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**self.button\_cal = Button(master, text="Browse Calendar", font=("bold", 9), relief="groove",**

**activebackground="red", command=self.cal1)**

**self.button\_cal.place(x=10, y=680)**

**self.show\_profile = Button(master, text="View User Info", font=("bold", 9), relief="groove",**

**activebackground="red", command=self.show\_info)**

**self.show\_profile.place(x=564, y=180)**

**self.logout = PhotoImage(file="Images\logout.png")**

**self.user\_logout = Button(master, image=self.logout, width=55, height=55,cursor="hand1", relief="flat",**

**activebackground="red", command=self.logOut)**

**self.user\_logout.place(x=659, y=659)**

**self.man = PhotoImage(file="Images\Man.png")**

**self.manphoto = Label(master, image=self.man)**

**self.manphoto.place(x=10, y=160)**

**self.fetch = Label(master, text="Fetch your Desire with Shahu! ", width=25, font=("bold", 16))**

**self.fetch.place(x=203, y=100)**

**self.s1 = PhotoImage(file="Images\s1.png")**

**self.search\_tools = Button(master, text="Search Tools", width=210, height=160, image=self.s1,**

**font=('arial', 16, "bold"), cursor="hand1", command=self.search,**

**activebackground="blue")**

**self.search\_tools.place(x=210, y=215)**

**self.search1 = Label(master, text="Search Tools ", font=("arial", 16,"bold"))**

**self.search1.place(x=240, y=395)**

**self.upload1 = PhotoImage(file="Images\yup1.png")**

**self.up = PhotoImage(file="Images\Vis3.png")**

**self.up = PhotoImage(file="Images\pload1.png")**

**self.upload\_tools = Button(master, text="Upload Tools", width=210, height=160, image=self.up,**

**font=('arial', 7, "bold"), cursor="hand1",command=self.upload,**

**activebackground="green",**

**activeforeground="red")**

**self.upload\_tools.place(x=438, y=215)**

**self.upload1 = Label(master, text="Upload Tools ", font=("arial", 16,"bold"))**

**self.upload1.place(x=475, y=395)**

**self.hire = PhotoImage(file="Images\hire1.png")**

**self.hire\_tools = Button(master, text="Hire Tools", width=210, height=160, image=self.hire,**

**font=('arial', 16, "bold"), cursor="hand1",command=self.hire1,**

**activebackground="cyan")**

**self.hire\_tools.place(x=210, y=445)**

**self.hire13 = Label(master, text="Hire Tools ", font=("arial", 16,"bold"))**

**self.hire13.place(x=257, y=623)**

**self.pay = PhotoImage(file="Images\paynow1.png")**

**self.pay\_tools = Button(master, text="Payment & Delivery", width=210, height=160, image=self.pay,**

**font=('arial', 16, "bold"), cursor="hand1",command=self.pay1,**

**activebackground="brown",**

**activeforeground="green")**

**self.pay\_tools.place(x=438, y=445)**

**self.pay13 = Label(master, text="Payment & Delivery ", font=("arial", 16,"bold"))**

**self.pay13.place(x=446, y=623)**

**# self.image\_tk = PhotoImage(self.select\_image())**

**# self.canvas.create\_image(0, 0, image=self.image\_tk)**

**def open\_terms(self):**

**os.startfile("Terms.txt")**

**def search(self):**

**self.master.withdraw()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = SearchTools(self.newWindow)**

**self.newWindow.geometry('650x550+450+140')**

**self.newWindow.title("Search Tools Form")**

**# This will redirect the user to upload tools GUI where after entering some details we can upload tools.**

**def upload(self):**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = uploadTools(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Upload Tools Form")**

**# This will redirect the user to Payment GUI where after entering some details we can generate invoice**

**# after returning the tool.**

**def pay1(self):**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = payTools(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Payment Tools Form")**

**# This will redirect the user to hire tools GUI where after entering some details we can sucessfully hire uploaded tools.**

**def hire1(self):**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = hireTools(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Hire Tools Form")**

**# This will show the user to Calendar which will be easy and convenient to know current time and date.**

**def cal1(self):**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = CalendarShow(self.newWindow)**

**self.newWindow.title("Calendar")**

**# def select\_image(self):**

**# This will redirect the user to search tools GUI where user can search the uploaded tools inorder to hire those tools.**

**def profile(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = SearchTools(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Search Tools Form")**

**# This will preview the instance information of newly registered user**

**def show\_info(self):**

**os.startfile("Text File Handling\ViewUserInfo.txt")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**# This logOut function will help user to logout him or her from software .**

**def logOut(self):**

**#print("This will logout you from user panel.")**

**tm.showwarning("Confirm LogOut",**

**"Are You sure want to LogOut from Shahu:The Ultimate Shared Power? ")**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = LoginFrame(self.newWindow)**

**self.newWindow.geometry('550x350+450+220')**

**self.newWindow.title("Shared Power Login Form")**

**def onRegister():**

**# json\_data = open(file\_directory).read()**

**tkinter.messagebox.showinfo("Successful!!","Boom!! Boom !! You are successfully registered!!")**

**def minimizeProgram():**

**#root.wm\_state('iconic')**

**root.withdraw()**

**#root.state("withdrawn")**

**def endProgram():**

**# top.quit()**

**tm.showinfo("Confirm Exit",**

**"Are You sure want to exit Shahu:The Ultimate Shared Power? " )**

**root.destroy()**

**# This will show the user to Calendar which will be easy and convenient to know current time and date.**

**class CalendarShow(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master)**

**self.label1 = Label(master, text="Year:")**

**self.label1.pack()**

**self.e1 = Entry(master)**

**self.e1.pack()**

**self.label2 = Label(master, text="Month:")**

**self.label2.pack()**

**self.e2 = Entry(master)**

**self.e2.pack()**

**self.button = Button(master, text="Show", command=self.cal)**

**self.button.pack()**

**master.title("Shahu-Calendar")**

**master.resizable(False, False)**

**master.overrideredirect(False)**

**def cal(self):**

**self.y = self.e1.get()**

**self.m = self.e2.get()**

**self.cal\_x = calendar.month(int(self.y), int(self.m), w=2, l=1)**

**#print(self.cal\_x)**

**self.cal\_out = Label( self.master,text=self.cal\_x, font=('courier', 12, 'bold'), bg="#0984e3")**

**self.cal\_out.pack(padx=3, pady=10)**

**'''**

**This is RegisterationFrame Class which will help the visitor user to registered into our software inorder to**

**use search , hire , upload, return tools and monthly invoice generation.**

**'''**

**class RegistrationFrame(Frame):**

**def \_\_init\_\_(self, master):**

**global entry\_fullname, \**

**entry\_Email, \**

**entry\_password, \**

**entry\_citizenshipno, \**

**country, \**

**var\_P, \**

**var\_nonP, \**

**register\_gender, gender\_value**

**self.master = master**

**self.frame = tk.Frame(master)**

**#self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17),relief="groove",activebackground="blue",command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_destroy = Button(master, text="X", width=2, font=("bold", 17) , relief="groove" , activebackground="red",command=endProgram)**

**self.button\_destroy.place(x=515, y=3)**

**self.label\_1 = Label(master, text="FullName",width=20,font=("bold", 13))**

**self.label\_1.place(x=80,y=130)**

**self.fullname=StringVar()**

**self.entry\_fullname = Entry(master,bd =5, textvariable=self.fullname)**

**self.entry\_fullname.place(x=240,y=130 ,width=180)**

**self.label\_2 = Label(master, text="Email",width=20,font=("bold", 13))**

**self.label\_2.place(x=68,y=180)**

**self.entry\_Email = Entry(master,bd =5)**

**self.entry\_Email.place(x=240,y=180,width=180)**

**self.label\_password = Label(master, text=" Password", width=20, font=("bold", 13))**

**self.label\_password.place(x=70, y=230)**

**self.entry\_password = Entry(master, bd=5,show="\*")**

**self.entry\_password.place(x=240, y=230, width=180)**

**self.label\_3 = Label(master, text="Gender",width=20,font=("bold", 13))**

**self.label\_3.place(x=70,y=280)**

**gender\_value = StringVar()**

**gender\_value.set(' ')**

**Radiobutton(master, text="Male", font=("bold", 10), padx=5 , variable=gender\_value, value="Male",**

**command=RegistrationFrame.selected\_gender).place(x=235, y=280)**

**Radiobutton(master, text="Female", variable=gender\_value, value="Female",**

**command=RegistrationFrame.selected\_gender).place(x=290, y=280)**

**'''**

**var = IntVar()**

**Radiobutton(master, text="Male",font=("bold", 10),padx = 5, variable=var, value=1).place(x=235,y=280)**

**Radiobutton(master, text="Female",font=("bold", 10),padx = 20, variable=var, value=2).place(x=290,y=280)**

**'''**

**self.label\_4 = Label(master, text="Country",width=20,font=("bold", 13))**

**self.label\_4.place(x=70,y=330)**

**list1 = ['Afghanistan','Algeria','Andorra','Angola','Antigua and Barbuda','Bangladesh', 'Thailand','Canada','India','UK','Nepal','Iceland','South Africa','Uganda','Maldives'];**

**country=StringVar()**

**country\_droplist =OptionMenu(master,country, \*list1, command=RegistrationFrame.countey\_selected)**

**country\_droplist.config(width=20)**

**country.set('Select your Country')**

**country\_droplist.place(x=240,y=330)**

**label\_q = Label(master, text=" CitizenNo.", width=20, font=("bold", 13))**

**label\_q.place(x=70, y=380)**

**self.entry\_citizenshipno = Entry(master, bd=5)**

**self.entry\_citizenshipno.place(x=240, y=380, width=180)**

**label\_4 = Label(master, text=" Applied Account",width=20,font=("bold", 13))**

**label\_4.place(x=85,y=430)**

**var\_nonP = IntVar()**

**Checkbutton(master, text="Non-Premium", variable=var\_nonP,command=RegistrationFrame.selected\_account).place(x=265,y=430)**

**var\_P = IntVar()**

**Checkbutton(master, text="Premium", variable=var\_P, command=RegistrationFrame.selected\_account).place(x=375,y=430)**

**self.registerButton=Button(master, text='Register',font=("bold", 11),width=20,bg='brown',fg='white' ,command =self.save\_info).place(x=208,y=462)**

**self.label\_5 = Button(master, text="Already registered Login with your info", width=65, font=('Times', 12,'underline'),cursor="hand1",activebackground="blue",command=self.command)**

**self.label\_5.place(x=0, y=500)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=80, y=542)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=20, y=23)**

**self.img1 = PhotoImage(file="Images\Regfinal1.png")**

**self.lab1 = Label(master, image=self.img1)**

**self.lab1.place(x=160,y=35)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def open\_terms(self):**

**os.startfile("Text File Handling\Terms.txt")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**@staticmethod**

**def selected\_gender(event=None):**

**global gender\_selected**

**gender\_selected = gender\_value.get()**

**#print(gender\_selected)**

**@staticmethod**

**def countey\_selected(event=None):**

**global country\_selected, name**

**country\_selected = country.get()**

**#print(country\_selected)**

**@staticmethod**

**def selected\_account(event=None):**

**global account\_selected2 , account\_selected1**

**account\_selected1 = var\_nonP.get()**

**account\_selected2 = var\_P.get()**

**account\_selected1=str(account\_selected1)**

**account\_selected2= str(account\_selected2)**

**#print(account\_selected2)**

**#print(account\_selected1)**

**def command(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = LoginFrame(self.newWindow)**

**self.newWindow.geometry('550x350+450+220')**

**self.newWindow.title("Shared Power Login Form")**

**#File handling by normal text format**

**def valid(self):**

**self.username = self.entry\_fullname.get()**

**self.password = self.entry\_password.get()**

**if len(self.username) == 0 and len(self.password)==0:**

**tm.showerror("Registration Error",**

**"Registration is unsucessful. Can be validation error or may be one or more field is empty.Please Register again with required Validation")**

**else:**

**print("Its ok Fine")**

**def save\_info(self):**

**self.name = self.entry\_fullname.get()**

**self.passw = self.entry\_password.get()**

**self.Username = self.entry\_Email.get()**

**self.Password = hashlib.sha1(self.entry\_password.get().encode()).hexdigest()**

**self.CNo = self.entry\_citizenshipno.get()**

**self.account\_selected1 = var\_nonP.get()**

**self.account\_selected2 = var\_P.get()**

**self.account\_selected1 = str(self.account\_selected1)**

**self.account\_selected2 = str(self.account\_selected2)**

**self.gender\_selected = gender\_value.get()**

**self.country\_selected = country.get()**

**self.Date = date.today()**

**if (len(self.name) == 0 and len(self.passw) == 0) and (len(self.passw) != 8):**

**tm.showerror("Registration Error",**

**"Registration is unsucessful. Can be validation error or may be one or more field is empty.Please Register again with required Validation")**

**else:**

**self.name = self.entry\_fullname.get()**

**self.passw = self.entry\_password.get()**

**self.Username = self.entry\_Email.get()**

**self.Password = hashlib.sha1(self.entry\_password.get().encode()).hexdigest()**

**self.CNo = self.entry\_citizenshipno.get()**

**self.account\_selected1 = var\_nonP.get()**

**self.account\_selected2 = var\_P.get()**

**self.account\_selected1 = str(self.account\_selected1)**

**self.account\_selected2 = str(self.account\_selected2)**

**self.gender\_selected = gender\_value.get()**

**self.country\_selected = country.get()**

**self.Date = date.today()**

**file = open("Text File Handling\database3.txt", "a")**

**file.write("\n")**

**file.write(" #####Registered User Information#####")**

**file.write("\n")**

**file.write("\n")**

**file.write("Name of user: ")**

**file.write(self.name)**

**file.write("\n")**

**file.write("UserName: ")**

**file.write(self.Username)**

**file.write("\n")**

**file.write("Password: ")**

**file.write(self.Password)**

**file.write("\n")**

**file.write("Account Selected: Non-Premium ")**

**file.write(self.account\_selected1)**

**file.write("\n")**

**file.write(" Premium ")**

**file.write(self.account\_selected2)**

**file.write("\n")**

**file.write("Gender: ")**

**file.write(self.gender\_selected)**

**file.write("\n")**

**file.write("Country: ")**

**file.write(self.country\_selected)**

**file.write("\n")**

**file.write("Citizenship Number: ")**

**file.write(self.CNo)**

**file.write("\n")**

**file.write("Registered Date: ")**

**file.write(str(self.Date))**

**file.write("\n")**

**file.close()**

**file1 = open("Text File Handling\ViewUserInfo.txt", "w")**

**file1.write("\n")**

**file1.write(" #####Registered User Information#####")**

**file1.write("\n")**

**file1.write("\n")**

**file1.write("Name of user: ")**

**file1.write(self.name)**

**file1.write("\n")**

**file1.write("UserName: ")**

**file1.write(self.Username)**

**file1.write("\n")**

**file1.write("Password: ")**

**file1.write(self.Password)**

**file1.write("\n")**

**file1.write("Account Selected: Non-Premium ")**

**file1.write(self.account\_selected1)**

**file1.write("\n")**

**file1.write(" Premium ")**

**file1.write(self.account\_selected2)**

**file1.write("\n")**

**file1.write("Gender: ")**

**file1.write(self.gender\_selected)**

**file1.write("\n")**

**file1.write("Country: ")**

**file1.write(self.country\_selected)**

**file1.write("\n")**

**file1.write("Citizenship Number: ")**

**file1.write(self.CNo)**

**file1.write("\n")**

**file1.write("Regsitered Date: ")**

**file1.write(str(self.Date))**

**file1.write("\n")**

**file1.close()**

**tm.showinfo("Successful!!",**

**"Boom!! Boom !! You are successfully registered!! Now yo will be redirected to LoginPage")**

**self.master.withdraw()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = LoginFrame(self.newWindow)**

**self.newWindow.geometry('550x350+450+220')**

**self.newWindow.title("Shared Power Login Form")**

**#File handling from JSON by using comparative understanding**

**'''**

**def onRegister(self):**

**self.name = self.entry\_fullname.get()**

**self.Username = self.entry\_Email.get()**

**self.Password = hashlib.sha1(self.entry\_password.get().encode()).hexdigest()**

**self.CNo = self.entry\_citizenshipno.get()**

**self.account\_selected1 = var\_nonP.get()**

**self.account\_selected2 = var\_P.get()**

**self.account\_selected1 = str(self.account\_selected1)**

**self.account\_selected2 = str(self.account\_selected2)**

**self.gender\_selected = gender\_value.get()**

**self.country\_selected = country.get()**

**# json\_data = open(file\_directory).read()**

**# tkinter.messagebox.showinfo("Successful!!","Boom!! Boom !! You are successfully registered!!")**

**# Button(root, text='Redirect ', font=("bold", 11), width=20, bg='brown', fg='white')**

**self.open\_db = json.load(open("database1.txt")) # Loads the json file as dictionary**

**self.profile = [{"Name": self.entry\_fullname.get(),**

**"Username": self.entry\_Email.get(),**

**"Password": hashlib.sha1(self.entry\_password.get().encode()).hexdigest(),**

**# The password from the password entry box is encoded with SHA1 in "Pass" value**

**"CNo": self.entry\_citizenshipno.get(),**

**"Gender": self.gender\_selected,**

**"Country": self.country\_selected,**

**"Account1": self.account\_selected1,**

**"Account2": self.account\_selected2,**

**"Own\_Tools": {},**

**"Hired\_Tools": {}**

**}]**

**self.open\_db[self.entry\_fullname.get()] = profile # Assigning the Name from entry box as a key**

**try:**

**json.dump(open\_db, open("database.txt",**

**'w')) # Saving the dictionary as json with "w" file method i.e, it overwrites the file**

**tm.showinfo("Successful!!", "Boom!! Boom !! You are successfully registered!!")**

**except:**

**tm.showinfo("UnSuccessful!!", "Invalid!! Something went wrong")**

**'''**

**'''**

**This is Upload Image class which give the user the function of to upload the desired image of respective tools**

**while uploading the tools details.**

**'''**

**class UploadImage(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.create\_widgets()**

**self.master.resizable(False, False)**

**def create\_widgets(self):**

**self.select = Button(self.master,text='Select Image', font=("arial", 13, "bold"), bg="green", fg='white',command=self.select\_image)**

**self.select.pack()**

**self.canvas = Canvas(self.master, width= 400, height=400, bg="grey")**

**self.canvas.pack()**

**self.store = Button(self.master, text='Store Image', font=("arial", 13, "bold"), bg="#e37b17", fg='white',**

**command=self.store\_image)**

**self.store.pack()**

**def select\_image(self):**

**global file\_path**

**file\_path = filedialog.askopenfilename()**

**des = Image.open(file\_path)**

**bg\_image = ImageTk.PhotoImage(des)**

**self.canvas.bg\_image = bg\_image**

**self.canvas.create\_image(200 , 200, image=self.canvas.bg\_image)**

**print(file\_path)**

**def store\_image(self):**

**#print("The selected tool image has been uploaded to our database.")**

**self.store = file\_path**

**file = open("Text File Handling\YUploadTools.txt", "a")**

**file.write("ToolImage: ")**

**file.write(self.store)**

**file.write("\n")**

**file.close()**

**tm.showinfo("Successfully Uploaded ToolImage!", "Your selected image is uploaded in ToolImage Database. !! Keep Exploring Shahu")**

**self.master.withdraw()**

**# print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = SearchTools(self.newWindow)**

**self.newWindow.geometry('650x550+450+140')**

**self.newWindow.title("Upload Tools Form")**

**'''**

**This is upload Tools class which enables the function of uploading Tool Details into our database which is carried out by**

**registered user so that the other registered user can hire the tools .**

**'''**

**class uploadTools(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master)**

**#self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17),relief="groove",activebackground="blue",command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_back = Button(master, text="X", width=2, font=("bold", 17) , relief="groove" , activebackground="red",command=self.back)**

**self.button\_back.place(x=515, y=3)**

**self.label\_name = Label(master, text="Tool Name",width=20,font=("arial", 17))**

**self.label\_name.place(x=30,y=165)**

**self.entry\_toolname = Entry(master,bd =5,font=("arial", 13))**

**self.entry\_toolname.place(x=280,y=165 ,width=200, height=38)**

**self.label\_tooldes = Label(master, text="Description",width=20,font=("arial", 17))**

**self.label\_tooldes.place(x=30,y=235)**

**self.entry\_tooldes = Entry(master,bd =5, font=("arial", 13))**

**self.entry\_tooldes.place(x=280,y=235,width=200, height=38)**

**self.label\_toolcondition = Label(master, text="Condition", width=20, font=("arial", 17))**

**self.label\_toolcondition.place(x=23, y=305)**

**self.entry\_toolcondition = Entry(master, bd=5, font=("arial", 13))**

**self.entry\_toolcondition.place(x=280, y=305, width=200, height=38)**

**self.label\_rate = Label(master, text="Tool Rate",width=20,font=("arial", 17))**

**self.label\_rate.place(x=30,y=375)**

**self.entry\_toolrate = Entry(master, bd=5, font=("arial", 13))**

**self.entry\_toolrate.place(x=280, y=375, width=90, height=38)**

**self.label\_fullrate = Label(master, text="Full Day",width=20,font=("arial", 10))**

**self.label\_fullrate.place(x=243,y=418)**

**self.entry\_toolrate2 = Entry(master, bd=5, font=("arial", 13))**

**self.entry\_toolrate2.place(x=390, y=375, width=90, height=38)**

**self.label\_halfrate = Label(master, text="Half Day",width=20,font=("arial", 10))**

**self.label\_halfrate.place(x=353,y=418)**

**Button(master, text='Upload Tools',font=("arial", 13,"bold"),width=15,bg='#e37b17',fg='white', command=self.upload\_info).place(x=298,y=464)**

**Button(master, text='Upload Image', font=("arial", 13, "bold"), bg="green", fg='white',**

**command=self.upload\_image1).place(x=58, y=464)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=80, y=512)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=20, y=23)**

**self.upload12 = PhotoImage(file="Images\pload12.png")**

**self.ph = Label(master, image=self.upload12)**

**self.ph.place(x=160,y=35)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def open\_terms(self):**

**os.startfile("Text File Handling\Terms.txt")**

**def upload\_image(self):**

**print("Lets begin to upload")**

**def upload\_image1(self):**

**tm.showwarning("Before Uploading TooolImage!",**

**"You should upload details of tool first then you need to upload Tool Image.\n If you have uploaded ToolDetails then Click OK to continue")**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**# self.master = master**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UploadImage(self.newWindow)**

**self.newWindow.title("Upload Tools Form")**

**# def select\_image(self):**

**# file\_path = filedialog.askopenfilename()**

**# return Image.open(file\_path)**

**def back(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+15')**

**self.newWindow.title("Shared Power Login Form")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**def upload\_info(self):**

**# global gender\_selected**

**#global country\_selected**

**#global account\_selected2, account\_selected1**

**#global name, Username, Password,CNo**

**self.nameTool = self.entry\_toolname.get()**

**self.ToolDescription = self.entry\_tooldes.get()**

**self.Toolcondition = self.entry\_toolcondition.get()**

**self.FullRate = self.entry\_toolrate2.get()**

**self.HalfRate = self.entry\_toolrate.get()**

**self.nameTool = self.entry\_toolname.get()**

**self.list1 = [self.nameTool, self.ToolDescription, self.Toolcondition, self.FullRate, self.HalfRate]**

**#print(self.list1)**

**self.Dict = {'Name of Tool:': self.nameTool, 'Tool Description:': self.ToolDescription, 'Tool Condition:':self.Toolcondition,'FullDayRate':self.FullRate,'HalfDayRate':self.HalfRate }**

**if len(self.nameTool) == 0 and len(self.ToolDescription) == 0 and len(self.Toolcondition) == 0 :**

**tm.showerror("Upload Tool Error",**

**"UploadTool is unsucessful. May be one or more field is empty.")**

**else:**

**self.ToolDescription = self.entry\_tooldes.get()**

**self.Toolcondition = self.entry\_toolcondition.get()**

**self.FullRate = self.entry\_toolrate2.get()**

**self.HalfRate = self.entry\_toolrate.get()**

**self.Date = date.today()**

**file = open("Text File Handling\YUploadTools.txt", "a")**

**file.write("\n")**

**file.write(" #####Registered User Upload Tools Info#####")**

**file.write("\n")**

**file.write("\n")**

**file.write("Name of Tool: ")**

**file.write(self.nameTool)**

**file.write("\n")**

**file.write("Tool Description: ")**

**file.write(self.ToolDescription)**

**file.write("\n")**

**file.write("Tool Condition: ")**

**file.write(self.Toolcondition)**

**file.write("\n")**

**file.write("Tool Rate: HalfDay ")**

**file.write(self.FullRate)**

**file.write("\n")**

**file.write(" Fullday ")**

**file.write(self.HalfRate)**

**file.write("\n")**

**file.write("Uploaded Date: ")**

**file.write(str(self.Date))**

**file.write("\n")**

**file.close()**

**file1 = open("Text File Handling\YUploadYes.txt", "a+")**

**#file1.write(str(self.list1))**

**file1.write(str(self.Dict))**

**file1.close()**

**#with open("uploadYes.txt", "w") as f:**

**#json.dump(self.Dict, f)**

**tm.showinfo("Successfully Uploaded Tool!", "Now your tool can be hired other Registered User . !! Keep Exploring Shahu")**

**'''**

**This is SearchTool Class which gives a function of searching uploaded tools inorder to get directed**

**to HireTool GUI for hiring a specific tool.**

**'''**

**class SearchTools(Frame):**

**def \_\_init\_\_(self, master):**

**global entry\_ToolName**

**global Searched\_Tool**

**self.master = master**

**self.frame = tk.Frame(master)**

**#self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17),relief="groove",activebackground="blue",command=self.minimizeProgram)**

**self.button\_minimize.place(x=591, y=2)**

**self.button\_back = Button(master, text="X", width=2, font=("bold", 17) , relief="groove" , activebackground="red",command=self.back)**

**self.button\_back.place(x=615, y=3)**

**self.label\_ToolName = Label(master, text="Tool Name",font=("arial", 22))**

**self.label\_ToolName.place(x=125,y=140)**

**Searched\_Tool = StringVar()**

**self.entry\_ToolName = Entry(master,bd =5, font=("arial", 14), textvariable=Searched\_Tool)**

**self.entry\_ToolName.place(x=296,y=140 ,width=230, height=40)**

**self.searched\_Tool = self.entry\_ToolName.get()**

**self.imgi = PhotoImage(file="Images\Vis3.png")**

**self.button\_vis = Button(master, image=self.imgi, font=("bold", 17),**

**activebackground="red", command=self.search\_results)**

**self.button\_vis.place(x=508, y=140)**

**Button(master, text='Wanna Hire This Tool', font=("arial", 13, "bold"), bg='#e37b17', fg='white', command=self.hire1).place(x=259,**

**y=470)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=140, y=512)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=20, y=10)**

**self.img1 = PhotoImage(file="Images\search3.png")**

**self.lab1 = Label(master, image=self.img1)**

**self.lab1.place(x=200,y=5)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**#def open\_terms(self):**

**#os.startfile("Terms.txt")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**def hire1(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**# self.master = master**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = hireTools(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Upload Tools Form")**

**# def select\_image(self):**

**# file\_path = filedialog.askopenfilename()**

**# return Image.open(file\_path)**

**def back(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+22')**

**self.newWindow.title("Shared Power Login Form")**

**def search\_results(self):**

**global Searched\_Tool**

**with open('Text File Handling\YUploadYes.txt', 'r') as f:**

**self.d = json.loads(f.read().replace("'", '"'))**

**#print (self.d['Name of Tool:'])**

**#self.a, self.b, self.c , self.d , self.e= self.d.split(',')**

**self.Searched\_Tool = self.entry\_ToolName.get()**

**#print(self.Searched\_Tool)**

**# print(self.a)**

**# print(self.b)**

**# print(self.c)**

**# print(self.d)**

**# print(self.e)**

**print(self.d['Name of Tool:'])**

**file = open("Text File Handling\YUploadTools.txt", "r")**

**self.searchTool = file.read()**

**self.yes=self.d['Tool Condition:']**

**if self.Searched\_Tool and self.Searched\_Tool != " " in self.searchTool:**

**self.show\_gui = Label(self.master, text=self.searchTool, font=('courier', 12, 'bold'), bg="#0984e3")**

**self.show\_gui.place(x=44, y=215)**

**#self.show\_gui.pack(padx=6, pady=200)**

**else:**

**tm.showerror("Invalid ToolName", "No Such Tool Is Uploaded By Any Registered User In Our Database")**

**def searchYes(self):**

**self.Searched\_Tool = self.entry\_ToolName.get()**

**return self.Searched\_Tool**

**'''**

**This is HireTools Class .This will enable a function of hiring a uploaded tool . Registered user can search tool**

**first and can be redirected to this page too for hiring the specific tool if he doesnt know the tool name.**

**'''**

**class hireTools(SearchTools):**

**def \_\_init\_\_(self, master):**

**global entry\_toolname**

**# self.showyes = ok.searched\_Tool**

**# self.var2 = master.var2**

**self.master = master**

**self.frame = tk.Frame(master)**

**# self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17), relief="groove",**

**activebackground="blue", command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_back = Button(master, text="X", width=2, font=("bold", 17), relief="groove", activebackground="red",**

**command=self.back)**

**self.button\_back.place(x=515, y=3)**

**self.label\_toolname = Label(master, text="Tool Name", width=20, font=("arial", 17))**

**self.label\_toolname.place(x=30, y=165)**

**with open('Text File Handling\YUploadYes.txt', 'r') as f:**

**self.d = json.loads(f.read().replace("'", '"'))**

**# self.controller.get\_page("searchTools").Searched\_Tool.get()**

**# label = tk.Label(self, text=VertexNumber)**

**# self.searchget = self.object1.searchYes()**

**self.entry\_toolname = Entry(master, bd=5, font=("arial", 13))**

**self.entry\_toolname.place(x=280, y=165, width=200, height=38)**

**#self.entry\_toolname.insert(0, "Wire Cutter")**

**self.entry\_toolname.insert(0, self.d['Name of Tool:'])**

**# self.entry\_toolname.insert(0, master.searched\_Tool)**

**# self.entry\_toolname.insert(0, "Wire Cutter")**

**self.label\_hireDate = Label(master, text="Hire Date", width=20, font=("arial", 17))**

**self.label\_hireDate.place(x=30, y=235)**

**self.Date = date.today()**

**self.entry\_hireDate = Entry(master, bd=5, font=("arial", 13))**

**self.entry\_hireDate.place(x=280, y=235, width=200, height=38)**

**self.entry\_hireDate.insert(0, self.Date)**

**self.label\_hireDays = Label(master, text="Hire Days", width=20, font=("arial", 17))**

**self.label\_hireDays.place(x=23, y=305)**

**self.entry\_hireDays = Entry(master, bd=5, font=("arial", 13))**

**self.entry\_hireDays.place(x=280, y=305, width=200, height=38)**

**self.entry\_hireDays.insert(0, " Max 3 Days")**

**# self.entry\_hireDays.bind("<FocusIn>", hireTools.clear\_hireDays)**

**self.label\_rate = Label(master, text="Tool Rate", width=20, font=("arial", 17))**

**self.label\_rate.place(x=30, y=375)**

**self.tool\_rate = Entry(master, bd=5, font=("arial", 13))**

**self.tool\_rate.place(x=280, y=375, width=90, height=38)**

**self.tool\_rate.insert(0, "234")**

**self.label\_fullrate = Label(master, text="Full Day", width=20, font=("arial", 10))**

**self.label\_fullrate.place(x=243, y=418)**

**self.tool\_rate2 = Entry(master, bd=5, font=("arial", 13))**

**self.tool\_rate2.place(x=390, y=375, width=90, height=38)**

**self.tool\_rate2.insert(0, "123")**

**self.label\_halfrate = Label(master, text="Half Day", width=20, font=("arial", 10))**

**self.label\_halfrate.place(x=353, y=418)**

**Button(master, text='Hire This Tool', font=("arial", 13, "bold"), width=18, bg='#1f3a93', fg='white'**

**, command=self.hired\_tools).place(x=210,**

**y=464)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9),**

**cursor="hand2")**

**self.label\_6.place(x=80, y=512)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=20, y=23)**

**self.hire1 = PhotoImage(file="Images\hiretool.png")**

**self.hire12 = Label(master, image=self.hire1)**

**self.hire12.place(x=173, y=43)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def open\_terms(self):**

**os.startfile("Text File Handling\Terms.txt")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**def hired\_tools(self):**

**# global gender\_selected**

**# global country\_selected**

**# global account\_selected2, account\_selected1**

**# global name, Username, Password,CNo**

**self.nameTool = self.entry\_toolname.get()**

**self.HireDate = self.entry\_hireDate.get()**

**self.HireDays = self.entry\_hireDays.get()**

**self.FullRate = self.tool\_rate2.get()**

**self.HalfRate = self.tool\_rate.get()**

**if len(self.nameTool) == 0 and len(self.HireDays) == 0 :**

**tm.showerror("Upload Tool Error",**

**"UploadTool is unsucessful. May be one or more field is empty.")**

**else:**

**self.nameTool = self.entry\_toolname.get()**

**self.HireDate = self.entry\_hireDate.get()**

**self.HireDays = self.entry\_hireDays.get()**

**self.FullRate = float(self.tool\_rate2.get())**

**self.HalfRate = float(self.tool\_rate.get())**

**self.InsurancePlus = self.HalfRate + 5.00**

**self.InsurancePlus2 = self.FullRate + 5.00**

**self.Date = date.today()**

**file = open("Text File Handling\hiredTools.txt", "a")**

**file.write("\n")**

**file.write(" #####Registered User Hired Tools Info#####")**

**file.write("\n")**

**file.write("\n")**

**file.write("Name of Tool: ")**

**file.write(self.nameTool)**

**file.write("\n")**

**file.write("Hired Date: ")**

**file.write(self.HireDate)**

**file.write("\n")**

**file.write("Hire Days: ")**

**file.write(self.HireDays)**

**file.write("\n")**

**file.write("Tool Rate: HalfDay $")**

**file.write(str(self.FullRate))**

**file.write("\n")**

**file.write(" Fullday $")**

**file.write(str(self.HalfRate))**

**file.write("\n")**

**file.write("\n")**

**file.close()**

**file1 = open("Text File Handling\displayReturn.txt", "a+")**

**file1.write(self.nameTool)**

**file1.write("\n")**

**file1.close()**

**file2 = open("Text File Handling\Invoice.txt", "w")**

**file2.write("\n")**

**file2.write(" ##### Auto generated Invoice Monthly #####")**

**file2.write("\n")**

**file2.write("\n")**

**file2.write("Name of Tool: ")**

**file2.write(self.nameTool)**

**file2.write("\n")**

**file2.write("Hired Date: ")**

**file2.write(self.HireDate)**

**file2.write("\n")**

**file2.write("Hire Days: ")**

**file2.write(self.HireDays)**

**file2.write("\n")**

**file2.write("Tool Rate: HalfDay $")**

**file2.write(str(self.FullRate))**

**file2.write("\n")**

**file2.write(" Fullday $")**

**file2.write(str(self.HalfRate))**

**file2.write("\n")**

**file2.write("InsuranceCharge $5.00")**

**file2.write("\n")**

**file2.write("Total: Fullday $")**

**file2.write(str(self.InsurancePlus))**

**file2.write("\n")**

**file2.write(" Halfday $")**

**file2.write(str(self.InsurancePlus2))**

**file2.write("\n")**

**file2.write("Generated Date: ")**

**file2.write(str(self.Date))**

**file2.write("\n")**

**file2.close()**

**tm.showinfo("Hired Tools Sucessful!! ",**

**"Your desired tool has been successfully added to your hired tool list. !!Keep Exploring Shahu!! ")**

**self.master.withdraw()**

**# print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+22')**

**self.newWindow.title("Shared Power Login Form")**

**def back(self):**

**# print(self.showyes)**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**# self.master = master**

**self.master.withdraw()**

**# print ('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+22')**

**self.newWindow.title("Shared Power Login Form")**

**'''**

**This is ReturnTool Class which enables the function of returning a hired tool which is most essential**

**in printing the invoice .**

**'''**

**class ReturnTools(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master)**

**#self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17), relief="groove",**

**activebackground="blue", command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_back = Button(master, text="X", width=2, font=("bold", 17), relief="groove", activebackground="red",**

**command=self.back)**

**self.button\_back.place(x=515, y=3)**

**self.listOfTools = Listbox(master, selectmode=EXTENDED, exportselection=0,font=("arial", 17, "bold")**

**,width=30, height=10,bg='#808e9b',fg='white',highlightcolor="green")**

**self.listOfTools.place(x=95, y=160)**

**self.data = []**

**with open("Text File Handling\displayReturn.txt", "r") as f:**

**for line in f:**

**self.data += line.splitlines()**

**# Create your listbox here.**

**for i in range(len(self.data)):**

**self.listOfTools.insert(i + 1, self.data[i])**

**Button(master, text='Return This Tool', font=("arial", 13, "bold"), width=18, bg='#1f3a93', fg='white'**

**, command=self.delete\_selected\_item).place(x=205,**

**y=464)**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9),**

**cursor="hand2")**

**self.label\_6.place(x=80, y=512)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\Returnlogo.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=10, y=31)**

**self.return1 = PhotoImage(file="Images\ReturnTools.png")**

**self.return12 = Label(master, image=self.return1)**

**self.return12.place(x=153,y=38)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def delete\_selected\_item(self):**

**tm.showwarning("Confirm ReturnTool",**

**"Are You sure want to return this tool ? If you return this tool , this tool will be deleted from "**

**"your hired tool database. ")**

**self.indexes = self.listOfTools.curselection()**

**for index in self.indexes:**

**self.listOfTools.delete(index)**

**def open\_terms(self):**

**os.startfile("Text File Handling\Terms.txt")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**def back(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+22')**

**self.newWindow.title("Shared Power Login Form")**

**'''**

**This is Return Tool and Payment Class which enables the function of this GUI whhich are returning a tool**

**and after returning tool the user can only preview invoice and inoreder to print the iinvoice it need to be last**

**day of month**

**'''**

**class payTools(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master)**

**#self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17),relief="groove",activebackground="blue",command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_back = Button(master, text="X", width=2, font=("bold", 17) , relief="groove" , activebackground="red",command=self.back)**

**self.button\_back.place(x=515, y=3)**

**self.returnTool = PhotoImage(file="Images\Tool2.png")**

**self.returnTool1 = Button(master, width=210, height=160, image=self.returnTool,**

**font=('arial', 16, "bold"), cursor="hand1",**

**activebackground="pink",**

**activeforeground="green", command=self.return1)**

**self.returnTool1.place(x=45, y=168)**

**self.ret13 = Label(master, text=" Return Tools ", font=("arial", 16, "bold"))**

**self.ret13.place(x=77, y=358)**

**#if draw.drawing == False:**

**#resetall.config(state=DISABLED)**

**#elif draw.drawing == True:**

**#resetall.config(state=NORMAL)**

**self.pay = PhotoImage(file="Images\invoice12.png")**

**self.pay\_tools = Button(master, text="Payment & Delivery", width=210, height=160, image=self.pay,**

**font=('arial', 16, "bold"), cursor="hand1",**

**activebackground="cyan",**

**activeforeground="green", command=self.print\_invoice)**

**self.pay\_tools.place(x=295, y=168)**

**self.pay13 = Label(master, text=" Print Invoice ", font=("arial", 16, "bold"))**

**self.pay13.place(x=318, y=358)**

**self.thankYou = Label(master, text="Thank You !! For Using Shahu ", width=30, font=("bold", 23))**

**self.thankYou.place(x=20, y=438)**

**'''**

**self.label\_name = Label(master, text="Tool Name",width=20,font=("bold", 13))**

**self.label\_name.place(x=80,y=130)**

**self.entry\_name = Entry(master,bd =5)**

**self.entry\_name.place(x=240,y=130 ,width=180)**

**self.label\_des = Label(master, text="Description",width=20,font=("bold", 13))**

**self.label\_des.place(x=68,y=180)**

**self.entry\_des = Entry(master,bd =5)**

**self.entry\_des.place(x=240,y=180,width=180)**

**self.label\_condition = Label(master, text="Condition", width=20, font=("bold", 13))**

**self.label\_condition.place(x=70, y=230)**

**self.entry\_condition = Entry(master, bd=5)**

**self.entry\_condition.place(x=240, y=230, width=180)**

**self.label\_rate = Label(master, text="Tool Rate",width=20,font=("bold", 13))**

**self.label\_rate.place(x=70,y=280)**

**self.entry\_rate = Entry(master, bd=5)**

**self.entry\_rate.place(x=240, y=280, width=180)**

**'''**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=60, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=80, y=512)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=20, y=21)**

**self.pay1 = PhotoImage(file="Images\Cash.png")**

**self.pay12 = Label(master, image=self.pay1)**

**self.pay12.place(x=194,y=27)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def print\_invoice(self):**

**self.date=date.today()**

**if self.date == self.date:**

**tm.showwarning("Something went wrong while generating invoice","Today is not the last of month \n.You cannot print the invoice but inorder to preview invoice click OK ")**

**os.startfile("Text File Handling\Invoice.txt")**

**file = open("Text File Handling\Invoice.txt", "r")**

**self.searchTool = file.read()**

**print(self.searchTool)**

**# if self.Searched\_Tool== " ":**

**# tm.showerror("Invalid ToolName", "No Such Tool Is Uploaded By Any Registered User In Our Database")**

**#self.show\_gui = Label(self.master, text=self.searchTool, font=('courier', 12, 'bold'), bg="#0984e3")**

**#self.show\_gui.place(x=18, y=220)**

**def open\_terms(self):**

**os.startfile("Text File Handling\Terms.txt")**

**def back(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = UserPanelFrame(self.newWindow)**

**self.newWindow.geometry('720x720+350+22')**

**self.newWindow.title("Shared Power Login Form")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**def return1(self):**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**# self.master = master**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = ReturnTools(self.newWindow)**

**self.newWindow.geometry('550x550+450+140')**

**self.newWindow.title("Upload Tools Form")**

**'''**

**This is Insurance Company Profile .Insurance Agent can login in our system by providing their valid**

**credentials inorder to start their session in our software.**

**'''**

**class InsuranceCompany(Frame):**

**def \_\_init\_\_(self, master):**

**self.master = master**

**self.frame = tk.Frame(master)**

**#self.label = Label(root, text="Registration form",width=20,fg='#1f3a93',bg = "#81cfe0 ",font=("Times", 30),borderwidth=3, relief="sunken").place(x=55,y=53)**

**self.button\_minimize = Button(master, text="\_", width=1, font=("bold", 17),relief="groove",activebackground="blue",command=self.minimizeProgram)**

**self.button\_minimize.place(x=491, y=2)**

**self.button\_back = Button(master, text="X", width=2, font=("bold", 17) , relief="groove" , activebackground="red",command=self.back)**

**self.button\_back.place(x=515, y=3)**

**self.title = Label(master, text="Welcome !! Explore Shahu ", width=30, font=("bold", 20))**

**self.title.place(x=55, y=55)**

**self.logout = PhotoImage(file="Images\logout.png")**

**self.user\_logout = Button(master, image=self.logout, width=55, height=55, cursor="hand1", relief="flat",**

**activebackground="red", command=self.logOut)**

**self.user\_logout.place(x=489, y=489)**

**self.title2 = Label(master, text=" Insurance Agent Profile", width=30, font=("bold", 17))**

**self.title2.place(x=91, y=125)**

**self.view = PhotoImage(file="Images\Viewinfo.png")**

**self.view\_info = Button(master, width=210, height=160, image=self.view,**

**font=('arial', 16, "bold"), cursor="hand1",**

**activebackground="cyan",**

**activeforeground="green", command=self.view\_info1)**

**self.view\_info.place(x=45, y=198)**

**self.view13 = Label(master, text=" View User Info ", font=("arial", 16, "bold"))**

**self.view13.place(x=70, y=408)**

**self.view2 = PhotoImage(file="Images\Viewtool.png")**

**self.view\_info1 = Button(master, width=210, height=160, image=self.view2,**

**font=('arial', 16, "bold"), cursor="hand1",**

**activebackground="cyan",**

**activeforeground="green", command=self.view\_info12)**

**self.view\_info1.place(x=295, y=198)**

**self.view132 = Label(master, text=" View All Uploaded Tool", font=("arial", 16, "bold"))**

**self.view132.place(x=270, y=408)**

**'''**

**self.entry\_condition = Entry(master, bd=5)**

**self.entry\_condition.place(x=240, y=230, width=180)**

**self.label\_rate = Label(master, text="Tool Rate",width=20,font=("bold", 13))**

**self.label\_rate.place(x=70,y=280)**

**self.entry\_rate = Entry(master, bd=5)**

**self.entry\_rate.place(x=240, y=280, width=180)**

**'''**

**self.label\_6 = Label(master, text="Copyright@ Developers of Shahu, 2019", width=55, font=('Helvetica', 9), cursor="hand2")**

**self.label\_6.place(x=80, y=512)**

**self.label\_6.bind('<Button-1>', SplashScreenFrame.open\_terms)**

**self.img2 = PhotoImage(file="Images\yup1.png")**

**self.lab2 = Label(master, image=self.img2)**

**self.lab2.place(x=5, y=21)**

**master.resizable(False, False)**

**master.overrideredirect(True)**

**def view\_info1(self):**

**os.startfile("Text File Handling\database3.txt")**

**def view\_info12(self):**

**os.startfile("Text File Handling\YUploadTools.txt")**

**# if self.Searched\_Tool== " ":**

**# tm.showerror("Invalid ToolName", "No Such Tool Is Uploaded By Any Registered User In Our Database")**

**#self.show\_gui = Label(self.master, text=self.searchTool, font=('courier', 12, 'bold'), bg="#0984e3")**

**#self.show\_gui.place(x=18, y=220)**

**def logOut(self):**

**#print("This will logout you from user panel.")**

**tm.showwarning("Confirm LogOut",**

**"Are You sure want to LogOut from Shahu:The Ultimate Shared Power? ")**

**self.master.withdraw()**

**#print('Button is pressed!')**

**# self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = LoginFrame(self.newWindow)**

**self.newWindow.geometry('550x350+450+220')**

**self.newWindow.title("Shared Power Login Form")**

**def open\_terms(self):**

**os.startfile("Text File Handling\Terms.txt")**

**def back(self):**

**tm.showwarning("Confirm Exit",**

**"Are You sure want to exit from Shahu:The Ultimate Shared Power? ")**

**# I need make windows itself destroy after clicking on this button and make other window appear in same position**

**#self.master = master**

**self.master.withdraw()**

**#print ('Button is pressed!')**

**#self.RegistrationFrame.destroy()**

**self.newWindow = tk.Toplevel(self.master)**

**self.app = LoginFrame(self.newWindow)**

**self.newWindow.geometry('550x350+450+220')**

**self.newWindow.title("Shared Power Login Form")**

**def minimizeProgram(self):**

**# root.wm\_state('iconic')**

**self.master.withdraw()**

**# root.state("withdrawn")**

**root = Tk()**

**root.geometry('600x370+340+190')**

**root.title("Shared Power Registration Form")**

**rf = SplashScreenFrame(root) #in this i need to change and try to implement all the frames in one coding**

**root.resizable(False, False)**

**root.overrideredirect(True)**

**#root.wm\_iconbitmap('Icon1.ico')**

**'''**

**Now this lines of codes are replaced by Kshitij Bajagain because ...these lines are not needed at all because**

**we are using photos for every windows through Calling Inside each classes(windows)**

**img1= PhotoImage(file="regfinal1.png")**

**lab1=Label(root, image=img1)**

**lab1.place(x=160,y=35)**

**img2= PhotoImage(file="yup1.png")**

**lab2=Label(root, image=img2)**

**lab2.place(x=20,y=23)**

**'''**

**root.mainloop()**