SBR MAHESHWARI VIDYAPEETH

PROJECT NAME: BANKING SYSTEM

SUBJECT: INFORMATICS PRACTICES

AISSCE- 2023

Submitted to

Ms. Naseem Khan

Submitted by

Tanmay Daga

ACKNOWLEDGEMENT

I, **TANMAY DAGA** of class XII –Science would sincerely like to thank **Ms. Naseem Khan, my Informatics Practices teacher** who has been a constant source of inspiration and guidance throughout the project work, acting as a guiding spirit behind the completion of this project. I am thankful to her for providing me the valuable suggestions and motivation for the completion of this project.

My sincere thanks also to our Principal **Mrs. Sarika Singh** who has encouraged in general the commerce students to be more analytical in solving the problems.

CERTIFICATE

This is to certify that **Tanmay Daga** of XII – Science has worked under my supervision by showing a great interest and completed his project to my total satisfaction.

The work and conduct of **Tanmay Daga** during the completion of his was excellent.

_____	_____	_____
Teacher's Signature	Principal's Signature	Examiner's Signature

_____

School Stamp

About Project

Bank Management System project is written in Python. The project folder contains a python script (main.py) and a folder with CSV files. This is a simple console based system which is very easy to understand and use. Talking about the system, it contains all the basic functions which include transferring amount, editing bank details, viewing bank details, viewing bank statistics, balance inquiry. In this mini project, there is a login system and it is being considered as admin can only access CSV files and can add and delete accounts.

Hardware and Software Requirements

Processor	11th Generation Intel Core i7-1165G7
Hard Disk	512 Gb Ssd
RAM	16 GB LPDDR4x
Standard I/O Devices	Keyboard, mouse ,Printer for Hard-Copy Reports etc ..
Software Requirement	Python: 3.9 and above Support Tools: MS WORD Operating System: Windows 10

TABLES

A) Users.Csv (100 rows of data)

username	password
Donna Blair	NRN18ONZ4SI
Kimberley Sanders	TBF24PRZ7WW
Paloma Hess	HGU83LRI6DX
Leah Barr	MFB78OVL5MA
Trevor Schmidt	GJN86RTI7RI
MacKensie Blanchard	MIR60NDI3UX
Zahir Mccarty	YIL68RQR1GW
Laura Charles	ATJ82XHC6RK
Sebastian Duffy	URD45UOF3YB
Aiko Tanner	XFR86VVC1CK
Xavier Leonard	VVT81BSB0PA
Jaden Mcintosh	UNJ18XHK0QO
Rina Pickett	WHI64ZQC5RT
Gil Cain	PLJ44GSI3CB
Myra Brooks	COB39QTO4OU
McKenzie Herring	TPU38SON7ID
Lana Goff	NTP44UEY1QF
Geoffrey Wong	GNH53UTA1WQ
Ulysses Bates	SGM07HNO8EF
Hiram Farmer	LJP84KYH4RE
Nola Osborne	UDH65HXR5PP
Brody Christensen	FRO45IZY8CZ
Willa Moran	QFG79BIQ7ZG
Kenyon Mcgee	MJK83JOD3CV
Emerald Lara	XRC37QUC8GF
Timon England	WBI68GUG2HN
Harriet Savage	RFS98BVU3RR
Francesca Collier	LNK68GMF4SS
Dorian Tyson	JYC36TPI4GF
Cherokee Fox	TPV62SVM1OB
Hiroko Landry	KLC89DJR2PW

username	password
Wendy Mills	ZTN03LDR4BF
Oprah Rasmussen	HDY43YHA3SW
Callum Mcdowell	CQV63OHI6TD
Noble Garrett	XBT23WHI9FB
Breanna Moody	RVC25DRN9VF
Vivien Ortiz	YJN43KWQ6BX
Hall Morse	LPT52QGU6KL
Cara Frank	ELH43CZC8II
Kiona Head	EKI56VQM7HT
Mannix Carlson	ZME73LMZ6FC
Slade Mendez	UYD31MLD5LL
Yvette Sullivan	FNQ57TTL5ET
Lev Phelps	QSK31EKU8KI
Dillon Moore	TWP40HLY5VI
Lee Burns	YCN77TOW7YW
Colby Fletcher	AIY35BEW2FE
Marah Hines	CMH78OOY2YG
Lane Rosales	DHT83NQV7SH
Burton Mccullough	MNB12NVH4VG
April Macdonald	ILU03ZAC5PS
Austin Neal	EKJ47WTD5DN
Mira West	CDG67CEQ0JS
Armand Love	VFS67IIE5KM
Vaughan Tyler	FTX27QDN5RC
Tatiana Mcdaniel	GNT84LMQ6KF
Josephine Christensen	IRF70GLF4XL
Castor Sears	EML44SRU6ML
Sophia Le	EJP54RIJ1OD
Steel Lucas	JGT78AQE8UP
Aurelia Delaney	WEA21CKF5CJ
Yolanda Edwards	ZRL83AHQ0HI

username	password
Preston Santiago	WJP53FXI6BO
Bernard Santos	JFD60GUK2NO
Dorothy Moon	DFI14WLU1VC
Hammett Douglas	DUP21GIB3RV
Peter Acevedo	LQV62NRT6QO
Lev Monroe	DQM71PKT3RP
Kelsey Beck	FER86ZHT9IU
Bert Robertson	MUC40FEQ4CF
Jane Decker	WJA08BDH8BC
Anika Dillard	EUG68NLV8HP
Regina Orr	XCX82SYJ2TS
Sydney Paul	JXD42PKY6VW
Keegan Hyde	NCD16YGT6CX
Cassidy Hopper	VNI26RND3QQ
Brady Guzman	NCA63KUR3IF
Brianna Mckay	PKY24PLI6EG
Christine Cameron	QWY85FTF1OE
Troy Walker	CCD57MEG3XT
Sharon Buck	TRU25PCG4AS
Otto Erickson	YFJ13UYN1DA
Kyla Harrington	CBE86HAI1BL
Wendy Rodriquez	SBG64VWB1TB
Kelly Cox	ECA51KJN2LU
Tatyana Burks	ZLM45SOC1WH
Karleigh Cotton	WGU04NPT7VC
Daquan Campbell	DOY22LXG7UE
Zahir Kerr	WFF39LDX2FB
Danielle Martin	SKI74GCK0EM
Craig Dickson	HQK95EQT4EN
Lenore Kim	UTH30EMD9EC
Meghan Dunlap	FDJ28JMX1VB
Hayden Christensen	WVZ30VHE8TB
Victoria Stout	IXD37LTJ4LQ
Owen Whitley	SDO33SUP2UG
Wade Miles	PTR55ZKE0QV
Leah Walton	VWD57QDD1KD
Marsden Bright	NBP13PNF8AL
Piper Holland	NMV39RLX7SH

B) Accounts.csv

Accounts

username	dob	address	email	balance	account_ope	account_type
Donna Blair	12.12.1997	citylight surat	helloworld@gmail.com	4133663	20.04.2022	Savings Account
Kimberley Sanders	22.08.1968	Ap #905-8105 Malesuada St.	ac.fermentum@aol.org	7917925	08.10.2021	Salary Account
Paloma Hess	09.11.1956	5228 Cras Ave	dictum@hotmail.couk	5082109	12.04.2023	Savings Account
Leah Barr	16.03.1954	588-9659 Cras Street	justo@yahoo.ca	8363443	26.07.2023	Salary Account
Trevor Schmidt	07.07.1975	Ap #129-5521 Ac Avenue	nulla@aol.org	5826665	16.01.2022	Savings Account
MacKensie Blanchard	29.08.1978	Ap #143-6669 Mi St.	ligula.consectetuer.rhoncus@outlook.couk	8295533	19.12.2022	Savings Account
Zahir Mccarty	26.09.1922	Ap #197-5074 Orci. Av.	eu.odio@google.ca	9052800	29.10.2021	Salary Account
Laura Charles	15.08.1947	764 Sem. Ave	sit.amet@protonmail.net	1044504	30.06.2022	Savings Account
Sebastian Duffy	13.10.1984	185-3679 Vivamus St.	elementum.dui@outlook.couk	7268035	21.05.2023	Savings Account
Aiko Tanner	28.07.1993	2439 Nulla. St.	ornare.fusce@protonmail.couk	2621497	07.02.2022	Savings Account
Xavier Leonard	21.12.1967	Ap #820-7455 Purus Ave	porttitor.vulputate@icloud.com	9710468	21.04.2023	Current Account
Jaden Mcintosh	07.12.1962	237 Id, Street	cum.sociis.natoque@outlook.edu	4834605	20.01.2022	Savings Account
Rina Pickett	27.12.1927	P.O. Box 464, 4630 Nulla Street	mauris@outlook.ca	2147047	30.01.2022	Savings Account
Gil Cain	31.01.1962	Ap #517-7817 Lectus Street	in.faucibus@yahoo.net	8287900	28.08.2023	Salary Account
Myra Brooks	09.01.1966	P.O. Box 628, 8057 Nullam St.	lobortis@icloud.com	4054680	17.05.2022	Current Account
McKenzie Herring	06.01.1992	P.O. Box 440, 8376 Eget Road	nunc.risus@hotmail.org	6410812	04.04.2023	Current Account
Lana Goff	11.07.1987	361-8712 Orci. Ave	integer.mollis@aol.com	6562146	25.07.2023	Current Account
Geoffrey Wong	25.03.1965	863-6946 Lectus Av.	cras.pellentesque@yahoo.org	3378477	09.08.2023	Current Account
Ulysses Bates	23.06.1955	4252 Vivamus Street	tristique.ac@icloud.net	6762884	14.01.2023	Savings Account
Hiram Farmer	22.04.1940	343-8333 Blandit Ave	egestas.blandit@hotmail.net	8406382	17.12.2022	Salary Account
Nola Osborne	10.05.1997	Ap #650-3317 At St.	nunc@icloud.net	6647053	20.06.2022	Salary Account
Brody Christensen	21.04.1939	P.O. Box 348, 8490 Vitae Ave	eu.placerat@yahoo.org	3384024	14.05.2022	Salary Account
Willa Moran	25.06.1957	7314 Molestie Street	porttitor@hotmail.couk	8583454	11.01.2022	Savings Account
Kenyon Mcgee	27.12.1955	5617 Sagittis. Rd.	porttitor.vulputate@outlook.net	8936190	19.10.2021	Current Account
Emerald Lara	18.05.1961	4032 Urna. Rd.	purus@hotmail.net	327791	27.12.2021	Savings Account
Timon England	16.01.1970	P.O. Box 599, 8818 Amet Road	magnis@google.net	6076369	29.04.2023	Salary Account
Harriet Savage	03.01.1964	P.O. Box 623, 5331 Ultrices Avenue	molestie.sed@yahoo.couk	6187633	24.03.2022	Salary Account
Francesca Collier	28.09.1965	2657 Velit. St.	lobortis@outlook.com	1133794	17.05.2022	Savings Account
Dorian Tyson	09.02.1974	Ap #252-2527 Gravida. St.	ornare@hotmail.com	6254130	17.01.2023	Savings Account

Cherokee Fox	30.07.1971	Ap #234-646 Elit Av.	eget.dictum@aol.org	285561	06.04.2022	Current Account
Hiroko Landry	18.05.1967	Ap #339-4671 Tincidunt Rd.	hendrerit@icloud.couk	8155147	27.07.2022	Salary Account
Wendy Mills	22.09.1994	Ap #716-762 Metus Rd.	ridiculus.mus@hotmail.org	4830181	03.08.2023	Salary Account
Oprah Rasmussen	19.06.1945	616-1695 Pellentesque St.	neque@protonmail.org	1870925	23.06.2023	Current Account
Callum Mcdowell	17.06.1945	P.O. Box 312, 5192 Velit Avenue	sociis.natoque@yahoo.com	7274592	08.08.2023	Savings Account
Noble Garrett	22.09.1928	3457 Id Street	molestie@google.com	4186431	07.11.2021	Savings Account
Breanna Moody	07.09.1942	534-6697 Curabitur Rd.	iaculis.enim@protonmail.net	6389648	08.12.2021	Savings Account
Vivien Ortiz	07.06.1991	234-5841 Libero. Ave	ridiculus.mus.proin@icloud.com	4696847	06.01.2023	Current Account
Hall Morse	07.03.1924	Ap #812-9945 Donec Rd.	fames@aol.edu	4034221	23.05.2022	Savings Account
Cara Frank	22.03.1938	P.O. Box 883, 2225 Cras Ave	et@aol.couk	4586075	28.02.2022	Current Account
Kiona Head	28.04.2003	Ap #250-1943 Venenatis Avenue	et.netus@aol.edu	6625817	06.02.2023	Salary Account
Mannix Carlson	24.10.1925	Ap #983-6816 A, Rd.	feugiat.non@aol.net	9671024	16.02.2023	Savings Account
Slade Mendez	15.04.1969	Ap #576-1424 Bibendum Av.	egestas.a.scelerisque@protonmail.com	1575789	05.01.2023	Salary Account
Yvette Sullivan	19.10.1961	P.O. Box 833, 1229 Neque. Ave	molestie.tortor.nibh@google.couk	5464527	08.11.2022	Salary Account
Lev Phelps	11.05.1999	Ap #739-9038 Mauris Avenue	consequat.lectus@hotmail.com	1259280	05.10.2021	Current Account
Dillon Moore	12.05.1929	P.O. Box 564, 5954 Tempor Street	dictum.eu.placerat@icloud.ca	3438334	02.05.2022	Current Account
Lee Burns	22.01.1957	Ap #200-6664 Eleifend St.	quis.pede@outlook.com	7102565	27.12.2022	Savings Account
Colby Fletcher	10.03.1931	8158 At Avenue	cursus@aol.ca	84357	04.04.2023	Current Account
Marah Hines	07.05.1979	288-4452 Ornare. Ave	in@google.net	1043802	17.04.2022	Salary Account
Lane Rosales	26.09.1950	763-2944 In, Rd.	maecenas.malesuada@hotmail.edu	3434357	27.05.2023	Savings Account
Burton Mccullough	13.02.1953	P.O. Box 328, 4027 Orci Avenue	fusce.dolor.quam@protonmail.org	7978402	20.09.2021	Savings Account
April Macdonald	16.04.1929	6043 Arcu. Rd.	pede.cum@outlook.net	1159391	09.06.2022	Current Account
Austin Neal	31.03.1964	8305 Ut Road	mauris.quis@outlook.net	8790797	29.08.2023	Salary Account
Mira West	19.10.1963	P.O. Box 789, 9894 Ridiculus Av.	donec.nibh@aol.net	5510479	13.08.2022	Current Account
Armand Love	27.05.1992	435-3111 Nec Rd.	donec@outlook.ca	5795972	10.02.2023	Current Account
Vaughan Tyler	17.11.1974	890-2850 Phasellus Rd.	viverra@yahoo.org	9893684	11.04.2023	Current Account
Tatiana Mcdaniel	09.04.1983	7441 Facilisis Road	placerat.eget.venenatis@yahoo.couk	8632036	09.11.2022	Savings Account
Josephine Christensen	03.06.1965	2680 Mauris. Road	felis@icloud.com	606747	24.11.2022	Salary Account
Castor Sears	24.01.1946	Ap #411-4207 A Avenue	vitae.nibh@aol.couk	7988997	07.07.2023	Current Account
Sophia Le	06.03.1942	426-4574 Diam Av.	ipsum.dolor@protonmail.com	6491645	04.06.2022	Salary Account

Steel Lucas	03.03.1977	830-3048 Augue Road	urna.nunc.quis@hotmail.org	152618	01.08.2023	Current Account
Aurelia Delaney	24.11.1996	996-382 Arcu Road	vel@protonmail.couk	6788937	18.04.2022	Savings Account
Yolanda Edwards	13.02.1976	297-6926 Molestie St.	et.arcu@google.couk	1841347	03.04.2023	Salary Account
Preston Santiago	13.03.1935	P.O. Box 264, 9997 Risus. St.	sollicitudin.orci@hotmail.couk	2277059	01.12.2021	Salary Account
Bernard Santos	20.06.1960	528-2335 Nullam Ave	imperdiet.dictum.magna@outlook.couk	3362304	08.05.2023	Savings Account
Dorothy Moon	03.06.1923	Ap #830-903 Donec St.	enim.consequat@yahoo.ca	9572846	17.08.2022	Savings Account
Hammett Douglas	29.08.1979	Ap #303-8168 Nisi St.	quis.arcu.vel@protonmail.ca	1970289	26.08.2022	Salary Account
Peter Acevedo	27.04.1927	940-5068 Purus Rd.	sit.amet@google.net	8111435	13.10.2021	Current Account
Lev Monroe	29.05.1933	Ap #396-8287 Porttitor St.	ornare.facilisis@yahoo.couk	7829094	27.09.2021	Salary Account
Kelsey Beck	26.06.1968	723-6713 Phasellus St.	neque.et@google.net	8249615	02.11.2021	Savings Account
Bert Robertson	28.05.2000	P.O. Box 740, 4903 Ultricies Street	suspendisse.aliquet.molestie@yahoo.edu	7350435	31.12.2021	Salary Account
Jane Decker	05.01.1949	9475 Felis. St.	luctus.lobortis@google.couk	278382	23.02.2022	Savings Account
Anika Dillard	04.06.1931	Ap #723-3537 Bibendum Road	nullam.nisl.maecenas@yahoo.edu	8108274	08.04.2022	Savings Account
Regina Orr	26.10.1943	Ap #156-7461 Donec Av.	condimentum.donec@hotmail.couk	585497	11.04.2022	Salary Account
Sydney Paul	21.09.1927	122-4562 Molestie St.	feugiat.non.lobortis@protonmail.org	3302748	31.10.2022	Current Account
Keegan Hyde	10.08.1992	P.O. Box 764, 6313 Risus. Ave	vel.faucibus.id@protonmail.couk	4270237	01.04.2023	Savings Account
Cassidy Hopper	13.12.1970	665-188 Consequat Road	erat.eget.ipsum@protonmail.couk	6682215	25.09.2021	Salary Account
Brady Guzman	24.01.1929	893-844 Et. St.	est@aol.ca	4622517	12.10.2022	Salary Account
Brianna Mckay	04.12.1951	Ap #982-8757 Libero Street	non.lobortis@yahoo.net	202194	04.08.2023	Savings Account
Christine Cameron	14.01.1973	995-3585 In St.	consectetuer.ipsum@outlook.com	9267007	21.09.2022	Salary Account
Troy Walker	08.12.1937	470-3011 Risus. Av.	sed.et.libero@google.ca	8496559	13.01.2023	Savings Account
Sharon Buck	20.05.1968	6402 Nisl. Ave	in.tempus.eu@hotmail.com	8887040	12.06.2023	Salary Account
Otto Erickson	11.08.2003	P.O. Box 900, 5963 Lectus. Ave	penatibus.et@yahoo.net	4970718	11.05.2023	Savings Account
Kyla Harrington	06.02.1945	693-5290 Consequat Rd.	cras.eu.tellus@aol.edu	5448994	17.03.2023	Salary Account
Wendy Rodriguez	26.10.1959	5333 Dignissim. Av.	at.velit@hotmail.org	3332336	13.09.2023	Salary Account
Kelly Cox	05.02.1969	951-5457 Convallis Street	a.aliquet.vel@yahoo.ca	897526	04.02.2023	Current Account
Tatyana Burks	03.06.1999	P.O. Box 553, 9623 Augue Avenue	vehicula.aliquet@yahoo.ca	2443450	25.09.2021	Salary Account
Karleigh Cotton	02.12.1999	P.O. Box 337, 611 Mauris Ave	est@outlook.couk	2035275	20.09.2021	Salary Account
Daquan Campbell	19.10.1956	Ap #113-235 At Rd.	ut@hotmail.com	237533	22.08.2022	Current Account
Zahir Kerr	13.07.1941	973-1159 Libero St.	nunc@protonmail.net	1043768	04.11.2021	Savings Account
Danielle Martin	26.02.1952	Ap #473-7675 Elit Av.	non.enim@aol.ca	4534805	09.03.2023	Salary Account
Craig Dickson	02.08.1966	Ap #799-169 Malesuada Rd.	consectetuer@yahoo.edu	7910106	25.02.2022	Savings Account
Lenore Kim	24.11.2000	7151 Non Rd.	quam.pellentesque@outlook.org	6336028	21.02.2022	Current Account
Meghan Dunlap	05.10.1979	Ap #725-8301 Dolor Av.	ac.feugiat@outlook.com	5661451	18.11.2022	Savings Account
Hayden Christensen	19.09.1963	Ap #348-5122 A Rd.	eleifend.non@google.edu	718465	20.06.2023	Current Account
Victoria Stout	16.10.1978	P.O. Box 220, 618 Sem Road	etiam.vestibulum.massa@outlook.com	4346168	27.10.2022	Current Account
Owen Whitley	01.02.1957	6611 Sodales Road	primis.in@icloud.net	8750535	17.10.2021	Current Account
Wade Miles	17.02.1973	307-3503 Ante. Road	velit@google.net	5896185	01.11.2022	Current Account
Leah Walton	14.07.1978	814-7669 Aliquam Rd.	adipiscing.lobortis@hotmail.net	5802837	22.02.2022	Salary Account
Marsden Bright	16.08.1983	Ap #842-7112 Ipsum. Rd.	malesuada.integer.id@aol.couk	864543	28.02.2022	Savings Account
Piper Holland	13.11.1958	689-9493 Proin Road	class@yahoo.edu	3848014	28.08.2023	Salary Account

C)Transactions.csv

from_user_id	to_user_id	amount	date_time
66	40	7981668	22.12.2014
84	40	71738613	23.03.2016
9	80	33074349	21.04.2020
26	68	55593769	25.11.2018
67	93	30592287	28.08.2016
82	93	8065190	04.12.2015
29	2	16403871	31.07.2020
7	6	71292706	24.12.2017
28	38	9451500	31.07.2023
66	89	61255536	09.11.2014
61	21	36842071	06.11.2021
40	48	45989638	11.02.2018
65	5	26973521	08.09.2020
92	86	64583046	17.03.2022
77	32	78734021	01.05.2016
20	33	34333899	26.07.2015
53	95	34219088	13.11.2022
44	73	36767120	21.04.2016
38	27	71209485	13.12.2015
42	51	1512944	21.10.2014
27	24	27038768	06.03.2018
9	25	44394413	21.03.2014
91	94	57795054	23.05.2019
15	1	73295660	16.10.2017
79	92	93187628	24.11.2016
15	67	22402068	18.08.2017
64	24	16585192	27.12.2016
53	31	30131752	24.08.2021
35	38	53505477	30.05.2017
49	64	7719729	11.09.2021
5	76	80483818	28.11.2013
4	71	43327818	18.11.2020
83	78	67083438	11.08.2020
26	38	13488007	21.09.2020
40	47	87003825	27.02.2020
37	79	36037478	03.01.2023
40	76	28496062	16.10.2019
53	33	193081	11.08.2021
66	43	40124746	01.03.2023
77	4	98206213	26.02.2022
12	99	79375507	02.08.2021
20	82	79015138	09.03.2022
52	1	14596442	01.03.2019
15	20	27764503	22.09.2022
64	43	34347315	01.12.2013
40	74	19754111	25.07.2017
67	14	71574524	18.11.2016
34	54	42067238	08.05.2017
1	50	82258097	10.10.2021
82	39	74612034	02.11.2014
65	1	67739320	29.06.2017
92	88	41145055	20.05.2017
77	93	66989428	03.12.2017
37	52	12942552	14.01.2020
91	56	93017596	20.08.2022
18	99	77609898	12.06.2023
77	83	72629913	23.12.2022
17	56	67748881	21.03.2014
36	81	55740382	20.04.2023
90	25	36675453	02.12.2016
27	30	71531269	13.07.2021
33	43	56574830	05.06.2014

Python Coding

Main.py

```
import datetime
import sys
import matplotlib.pyplot as plt
import pandas as pd

UserId = None  # To be accessible by all functions
## PATHS ----------------
PATH_ACCOUNTS = ".\\CSV FILES\\Accounts.csv"
PATH_TRANSACTIONS = ".\\CSV FILES\\Transactions.csv"
PATH_USERS = ".\\CSV FILES\\Users.csv"

df = pd.read_csv(PATH_ACCOUNTS)


def main():
    global UserId
    print("---------------------------------------")
    print("Welcome to Central Banking System")
    print("---------------------------------------")

    isAuthenticated = authentication()
    if isAuthenticated[0]:
        UserId = isAuthenticated[1]
    Choice()

    exitProgram()


def authentication():
    """
    Function authenticates the user i.e. if it has entry in Users.csv with correct password
    :return: [bool,int/None]
    """

    def auth():
        """
        Helper function for authentication
        created to not again and again call userId,password input
        :return:[bool,int/None]
        """
        userId = input("Please enter your user name:")
        password = input("Please enter your password: ")
        df_Users = pd.read_csv(PATH_USERS)  # Only used for verifying password


        """
        Pandas isin() method is used to filter data frames. isin() method helps in selecting rows with having a
particular(or Multiple) value in a particular column.
        searchResult = df["username"].isin([UserId])->Dataframe [(0,True),(1,False),(2,False).....]
        using df over it only gets true values
        """
        searchResult = df_Users[df_Users["username"].isin([userId])]

        if not bool(searchResult.empty):
```

```
            if (searchResult["password"] == password).at[0]: # searchResult["password"] == password returns
dataframe with bool value
                return [True, searchResult.index[0]]
            return [False, None]
        return [False, None]

    for i in range(3):
        isauth = auth()
        if isauth[0]: return isauth
        print("Wrong Credentials!!!")
        if i == 2: break # To prevent want to try to run after 3 tries
        wanttoRetry = input("Do you want to retry?\nIf yes press 1 else 2")
        if wanttoRetry == "1":
            continue
        else:
            exitProgram()
    print("You have exceeded maximum no of tries")
    exitProgram()


def Choice():
    """
    Function to be called in main to get user choice
    main event loop
    :return: None
    """

    def getChoice():
        for i in range(3):
            choice = input(
                "\n\nPress 1 to view balance\nPress 2 to transact amount\nPress 3 to edit details\nPress 4 to see
"
                "bank statistics\nPress 5 to view details\nPress 6 to exit\n")
            if choice in ["1", "2", "3", "4", "5", "6"]:
                return [True, choice]
            if i == 2: break  # To prevent want to try to run after 3 tries
            # The line runs if not a valid choice
            wanttoRetry = input("Do you want to retry?\nIf yes press 1 else 2")
            if wanttoRetry == "1":
                continue
            else:
                break
        print("You have exceeded maximum no of tries")
        return [False, None]

    while True:
        choice = getChoice()
        if choice[1] == "1":
            print(f"Your balance is {getBalance()}")
        elif choice[1] == "2":
            transact_amount()
        elif choice[1] == "3":
            editDetails()
        elif choice[1] == "4":
            showGraphs()
        elif choice[1] == "5":
            printDetails()
        elif choice[1] == "6":
            exitProgram()
        else:
            print("Wrong Input!! Please Try again")
```

```
# -------------------------------------Choice 1---------------------------------
def getBalance():
    """Returns balance of user"""
    return df.at[UserId, "balance"]


# -------------------------------------Choice 2---------------------------------
def transact_amount():
    """
    transfer amount from one user to other
    :return:
    """
    user_id_to_transfer = int(input("Please enter userid to whom you want to transfer amount"))

    if user_id_to_transfer in df.index.values: # df.index.values -> 0,1,2,3,4,5 .....
        amount_to_transfer = int(input("Please enter the amount to be transferred"))
        if 0 < amount_to_transfer:
            userBankBalance = getBalance()
            if userBankBalance < amount_to_transfer:
                print("Not enough bank balance!!")
            else:
                df.at[UserId, "balance"] = userBankBalance - amount_to_transfer
                df.at[user_id_to_transfer, "balance"] += amount_to_transfer
                df.to_csv(PATH_ACCOUNTS, index=False)

                dt = pd.read_csv(PATH_TRANSACTIONS)

                # adding transaction to last row
                dt.iloc[-1] = [UserId, user_id_to_transfer, amount_to_transfer, datetime.datetime.now()]

                # Pandas reset_index() is a method to reset index of a Data Frame.
                # reset_index() method sets a list of integer ranging from 0 to length of data as index.
                dt = dt.reset_index(drop=True) #
                dt.to_csv((PATH_TRANSACTIONS), index=False)
                print("Transaction Successful")
        else:
            print("Invalid Amount!!")


    else:
        print("No user with such userid")


# -------------------------------------Choice 3---------------------------------
def editDetails():
    """
    Helps to edit user details
    """

    while True:
        choice = input(
            "Press 1 to change dob\nPress 2 to change address\nPress 3 to change email\nPress 4 to go to
upper menu\n")
        if choice == "1":

            # DOB -  string -- dd.mm.yyyy
            newDob = input(f"Please enter dob in format dd.mm.yyyy")
            try:
                year = int(newDob[6:])
                month = int(newDob[3:5])
                day = int(newDob[0:2])
                if isValidDate(year, month, day):
                    df.at[UserId, "dob"] = newDob
```

```
                saveAccounts()
                print("Date of Birth changed Successfully")
            except Exception:
                print("Invalid Date!!")
                return

        elif choice == "2":
            newAddress = input("Please enter new Address")
            df.at[UserId, "address"] = newAddress
            saveAccounts()
            print("Address Changed Successful")
        elif choice == "3":
            newEmail = input("Please enter new email")
            if "@" in newEmail:
                df.at[UserId, "email"] = newEmail
                saveAccounts()
                print("Email Changed Successful")
            else:
                print("Invalid Email!!")
        elif choice == "4":
            break
        else:
            print("Invalid Choice!!")


def isValidDate(year, month, date) -> bool:
    try:
        datevalid = datetime.datetime(year, month, date)

        # checking if age is less than 100 and greater than 18
        if datetime.datetime(datetime.datetime.today().year - 100, datetime.datetime.today().month,
                             datetime.datetime.today().day) < datevalid < datetime.datetime(
            datetime.datetime.today().year - 18, datetime.datetime.today().month, datetime.datetime.today().day):
            return True

        else:
            raise Exception
    except Exception:
        print("Invalid Date!!")
        return False


# -------------------------------------Choice 4---------------------------------
def showGraphs():

    def showTypesOfAccountGraph():

        plt.title(label="Different types of account")

        # # -- Types of account pie chart
        listOfTypesOfAccounts = df["account_type"].tolist()  # if not list then error
        noOfSavingsAccounts = listOfTypesOfAccounts.count("Savings Account")
        noOfCurrentAccounts = listOfTypesOfAccounts.count("Current Account")
        noOfSalaryAccounts = listOfTypesOfAccounts.count("Salary Account")
        plt.pie([noOfSalaryAccounts, noOfCurrentAccounts, noOfSavingsAccounts],
                labels=["Salary Account", "CurrentAccount", "Savings Account"])
        plt.show()

    def showdifferentAgedPeopleGraph():

        plt.title(label="Different Age Group People")
        dobs = df["dob"].tolist()
        ages = []
        todayYear = datetime.datetime.today().year
```

```
        ## dobs to datetime object
        for i in dobs:
            ages.append(todayYear - int(i[6:]))
        """
        Three age group:
        18-30,
        30:50,
        50 and above
        """
        age18to30 = 0
        age30to50 = 0
        age50andabove = 0
        for x in ages:
            if x <= 30:
                age18to30 += 1
            elif 30 < x < 50:
                age30to50 += 1
            else:
                age50andabove += 1

        plt.hist(ages)


        plt.show()

    def showAccountsOpenedLastYearGraph():
        plt.title(label="Accounts Opened")
        accountsOpened = df["account_open_date"]
        year_account_opened_2021 = 0
        year_account_opened_2022 = 0
        year_account_opened_2023 = 0
        for i in accountsOpened:
            if i[6:] == "2021": year_account_opened_2021 += 1
            if i[6:] == "2022": year_account_opened_2022 += 1
            if i[6:] == "2023": year_account_opened_2023 += 1
        plt.barh(["2021", "2022", "2023"],
                 [year_account_opened_2021, year_account_opened_2022, year_account_opened_2023],
color="yellow")

        plt.show()
    def showLast10YearTransaction():
        plt.title(label="Transactions in last 10 years")
        dt = pd.read_csv(PATH_TRANSACTIONS)

        # date format - string - dd.mm.yyyy
        year10transactions = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        for i in dt["date_time"]:
            if i[6:] == "2014":
                year10transactions[0] += 1
            elif i[6:] == "2015":
                year10transactions[1] += 1
            elif i[6:] == "2016":
                year10transactions[2] += 1
            elif i[6:] == "2017":
                year10transactions[3] += 1
            elif i[6:] == "2018":
                year10transactions[4] += 1
            elif i[6:] == "2019":
                year10transactions[5] += 1
            elif i[6:] == "2020":
                year10transactions[6] += 1
            elif i[6:] == "2021":
                year10transactions[7] += 1
            elif i[6:] == "2022":
                year10transactions[8] += 1
            elif i[6:] == "2023":
```

```
            year10transactions[9] += 1

        plt.plot(["2014", "2015", "2016", "2017", "2018", "2019", "2020", "2021", "2022", "2023"],
year10transactions)
        plt.xlabel(xlabel="Years")
        plt.ylabel(ylabel="Number of Transactions")
        plt.xticks(rotation=65)
        plt.show()




    def showPeoplewithdifferentemailsubdomainGraph():
        plt.title(label="People with different emails")
        emailsList = df["email"]
        differentemailsdomainy = [0, 0, 0, 0, 0, 0, 0]
        differentemailsdomainx = ["hotmail", "yahoo", "google", "aol", "icloud", "protonmail", "others"]
        for i in emailsList:
            if "hotmail" in i:
                differentemailsdomainy[0] += 1
            elif "yahoo" in i:
                differentemailsdomainy[1] += 1
            elif "google" in i:
                differentemailsdomainy[2] += 1
            elif "aol" in i:
                differentemailsdomainy[3] += 1
            elif "icloud" in i:
                differentemailsdomainy[4] += 1
            elif "protonmail" in i:
                differentemailsdomainy[5] += 1
            else:
                differentemailsdomainy[6] += 1

        plt.bar(differentemailsdomainx, differentemailsdomainy, color="cyan")

        plt.xticks(rotation=45)

        plt.show()


    while True:
        choice = input(
            "Press 1 to show different account types ratio\nPress 2 to show different aged people ratio having
account in bank\nPress 3 ratio of accounts opened in recent years\nPress 4 to show last 10 years of
bank\nPress 5 to show number of people with different email subdomains\nPress 6 to go to upper menu")
        if choice == "1":
            showTypesOfAccountGraph()
        elif choice == "2":
            showdifferentAgedPeopleGraph()
        elif choice == "3":
            showAccountsOpenedLastYearGraph()
        elif choice == "4":
            showLast10YearTransaction()
        elif choice == "5":
            showPeoplewithdifferentemailsubdomainGraph()
        elif choice == "6":
            break
        else:
            print("Invalid Choice")

def printDetails():
    with pd.option_context('display.max_rows', 5, 'display.max_columns', None):
        data = df.loc[[UserId], ["username", "dob", "address", "email", "account_open_date", "account_type"]]
        print(data.to_string(index=False))
```

```
    # ----------------------------------Choice 6-------------------------------------



def exitProgram():
    print("--------------------------------------")
    print("Thank you for using Central Banking System")
    print("--------------------------------------")
    sys.exit(0)


def saveAccounts():
    df.to_csv(PATH_ACCOUNTS, index=False)


main()
```

Output Screen

```
-------------------------------------
Welcome to Central Banking System
-------------------------------------
Please enter your user name:Donna Blair
Please enter your password: NRN18ONZ4SI


Press 1 to view balance
Press 2 to transact amount
Press 3 to edit details
Press 4 to see bank statistics
Press 5 to view details
Press 6 to exit
```

On inputing 1

```
1
Your balance is 4179284
```

On inputing 2

On inputing valid user id and amount

```
2
Please enter userid to whom you want to transfer amount45
Please enter the amount to be transferred45621
Transaction Successful
```

On inputing 3 invalid userid

```
2
Please enter userid to whom you want to transfer amount97843
No user with such userid
```

On inputing invalid bank balance

```
2
Please enter userid to whom you want to transfer amount45
Please enter the amount to be transferred29437893274
Not enough bank balance!!
```

On inputing 3

```
3
Press 1 to change dob
Press 2 to change address
Press 3 to change email
Press 4 to go to upper menu
```

On inputing 1

On inputing valid date

```
1
Please enter dob in format dd.mm.yyyy12.12.1997
Date of Birth changed Successfully
```

On inputing invalid date

```
1
Please enter dob in format dd.mm.yyyy45.34.21
Invalid Date!!
```

On inputing 2

```
Please enter new Addresscitylight surat
Address Changed Successful
```

On inputing 3

On inputing valid email

```
Please enter new emailhelloworld@gmail.com
Email Changed Successful
```

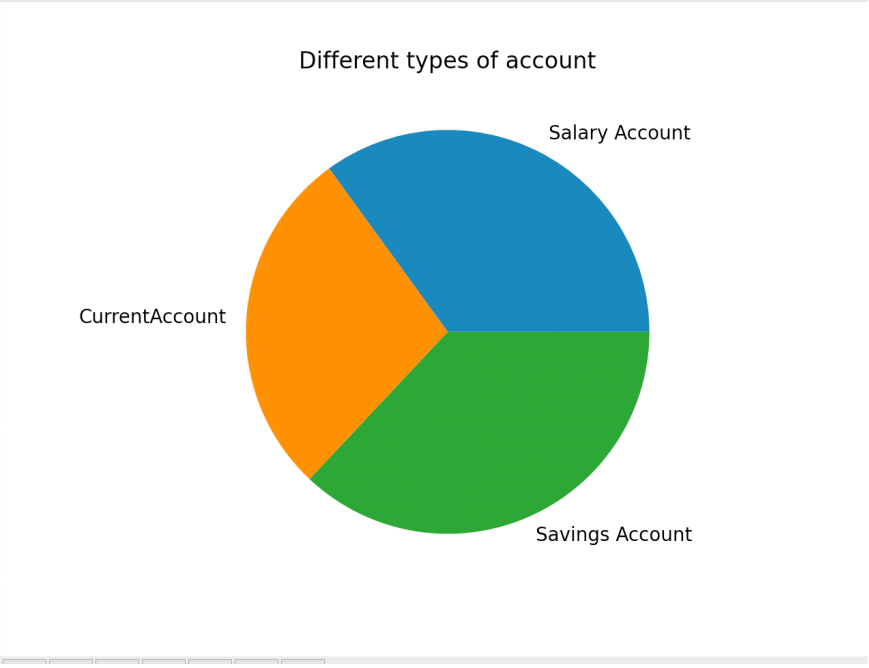
On inputing invalid date

```
Please enter new emailhellowlakfjds
Invalid Email!!
```

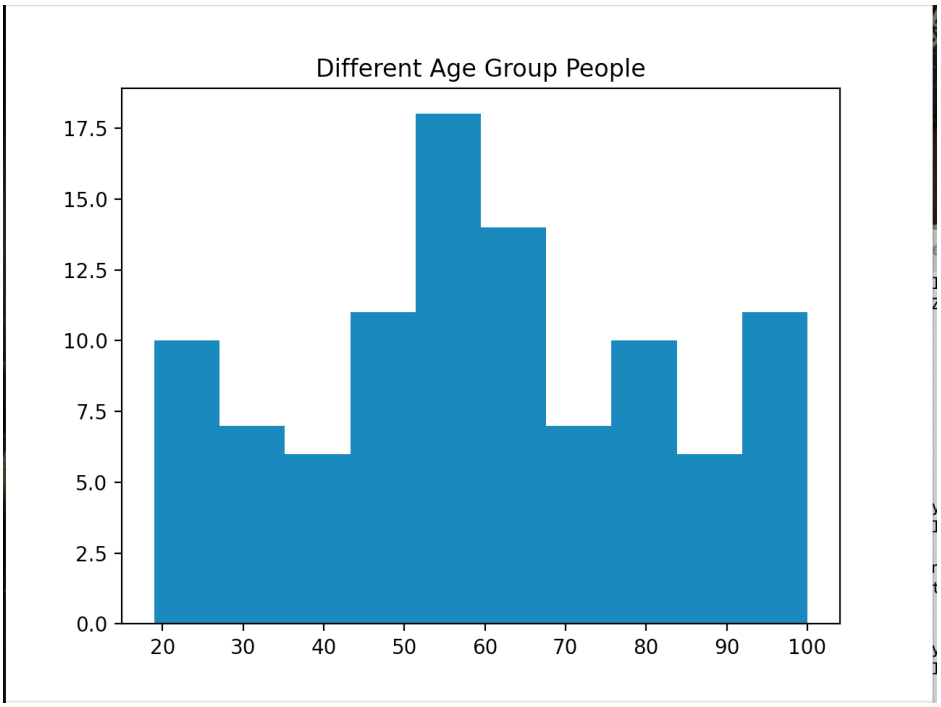
On inputing 4

```
Press 1 to show different account types ratio
Press 2 to show different aged people ratio having account in bank
Press 3 ratio of accounts opened in recent years
Press 4 to show last 10 years of bank
Press 5 to show number of people with different email subdomains
Press 6 to go to upper menu
```

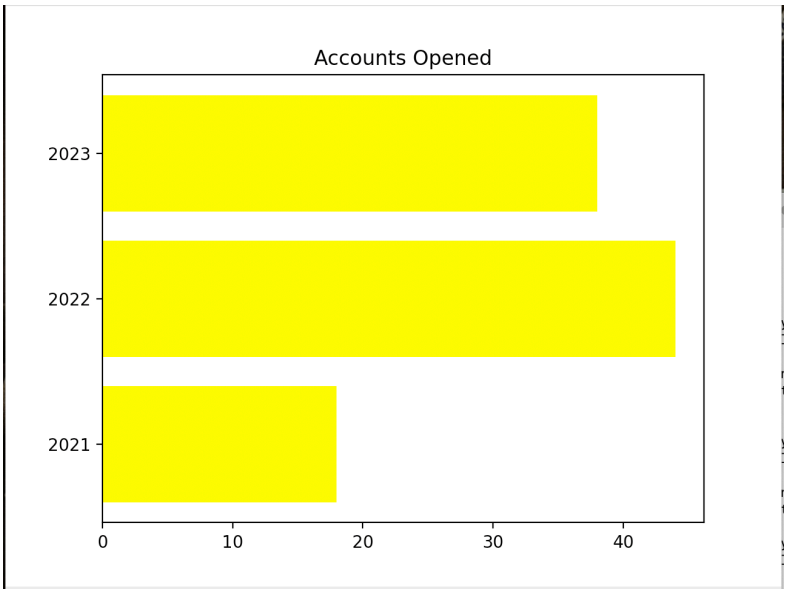
On inputing 1



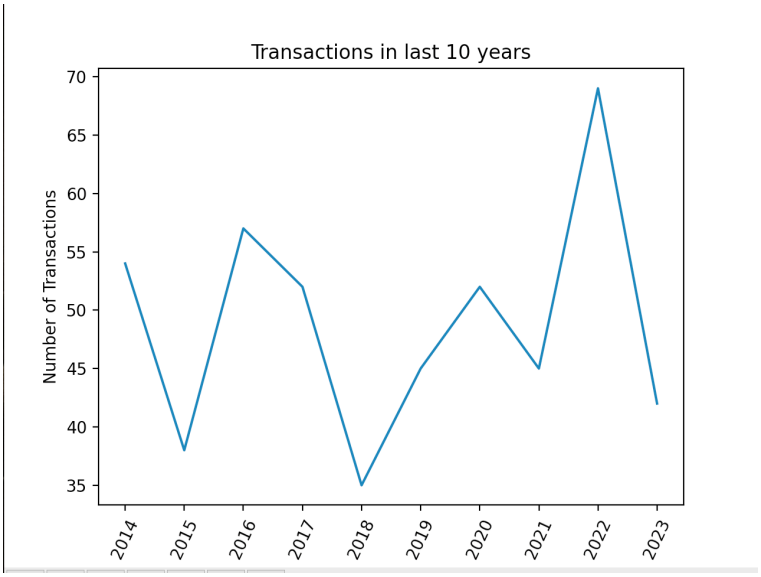
On inputing 2



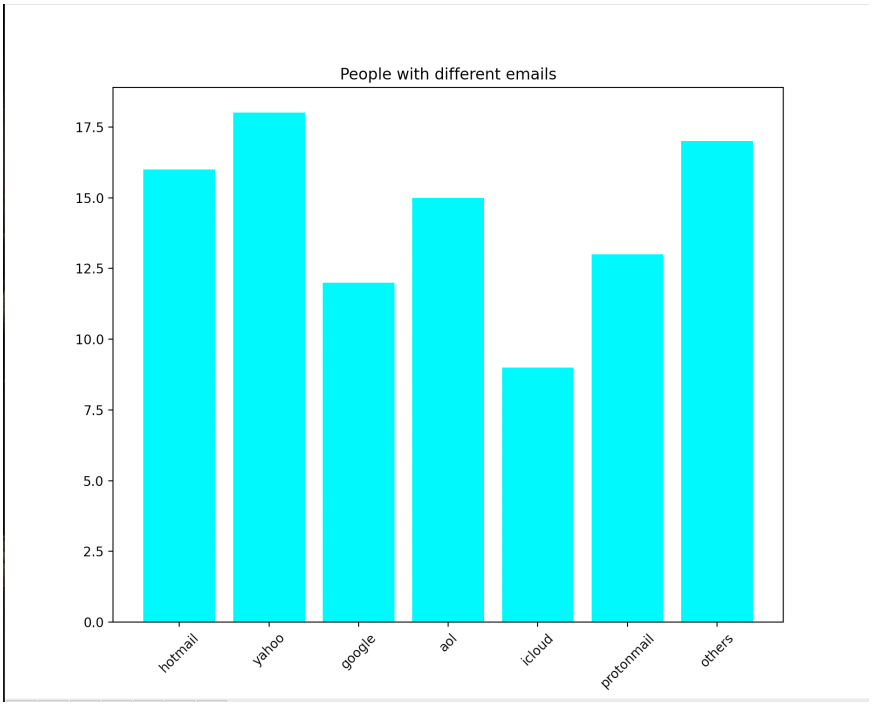
On inputing 3



On inputing 4



On inputing 5



On inputing 5

```
5
   username        dob         address                email account_open_date    account_type
Donna Blair 12.12.1997 citylight surat helloworld@gmail.com        20.04.2022 Savings Account
```

On inputing 6

```
6
--------------------------------------
Thank you for using Central Banking System
--------------------------------------
```

Bibliography

1. Informatics Practices[NCERT]
2. Informatics Practices[Sumita Arora]
3. https://generatedata.com for generating random data
4. https://www.geeksforgeeks.org