PERSONAL ADDRESS ENTRY BOOK

Project Link
https://github.com/GVicky98/Personal-Address-EntryBook

Project by

G. VIGNESH

Date

December 6, 2022

Course title

CRUD Project in Python using MySQL

Teacher's Name

Mrs. Meenaatchi

TABLE OF CONTENTS

ABSTRACT

Personal Address Entry book is a CRUD app concept developed using Core Python and MySQL Database connectivity. The CRUD app is built in a way that it uses the Python Interface as Front-end and the MySQL as Back-end. This app let the users to Signup or Login to the account. After login it lets the users to Create and store the database entry, Read, and view the database entries, Update the entries and Delete the entries in the MySQL database. Finally, upon finishing all the activities the user may either Logout or Exit the app.

INTRODUCTION

WHAT IS A CRUD APP?

CRUD apps are the user interface that we use to interact with databases through APIs. It is a specific type of application that supports the four basic operations: *Create, Read, Update, Delete.* Broadly, CRUD apps consist of the database, the user interface, and the APIs.

Modern web applications have a user perform at least one of the following operations on a database – creating (new record), read/view (existing records), update (existing records), or delete (a record). Or shortly, CRUD. If you have worked with a database, you have most likely worked with CRUD without realizing.

CRUD apps are used daily by several businesses and organizations to maintain their day-to-day workflows. HR uses CRUD apps to manage staff records and track keeping of employee leaves, attendance. Sales uses CRUD to manage lean information, customer success uses CRUD apps to resolve customer tickets, and so on. Have a look at blogging sites. You as a user can Create/Publish a post, Read/Show your post, Update/Edit your post and Delete your post. Blogs are classic CRUD apps.

SOFTWARE USED:

- Python v3.11
- MySQL v8.0

HARDWARE USED:

- Lenovo Legion5
- RAM: 8Gb
- ROM: 1Tb HDD & 256 Gb SDD
- Processor: AMD Ryzen 5 4600H

IDE USED:

- PyCharm Community Edition 2022.1.3
- MySQL Workbench 8.0 CE

MAJOR CONCEPTS USED:

- Modules
- File handling
- Regular Expressions
- Exceptional handling
- OOPs concept
- Python – MySQL Connectivity

APP OVERVIEW

In this CRUD application Python is used a Front-end tool. User can use and access the app by entering the input in the Python Interface. MySQL is used as a Back-end tool. The database will be retrieved from the MySQL database to the Python. Initially if the user has already created the account, then they can login to the app. Otherwise the user must create an account, in order to continue to the app. After Login process, the user can Create, Update, Read/View, and Delete the Entries.

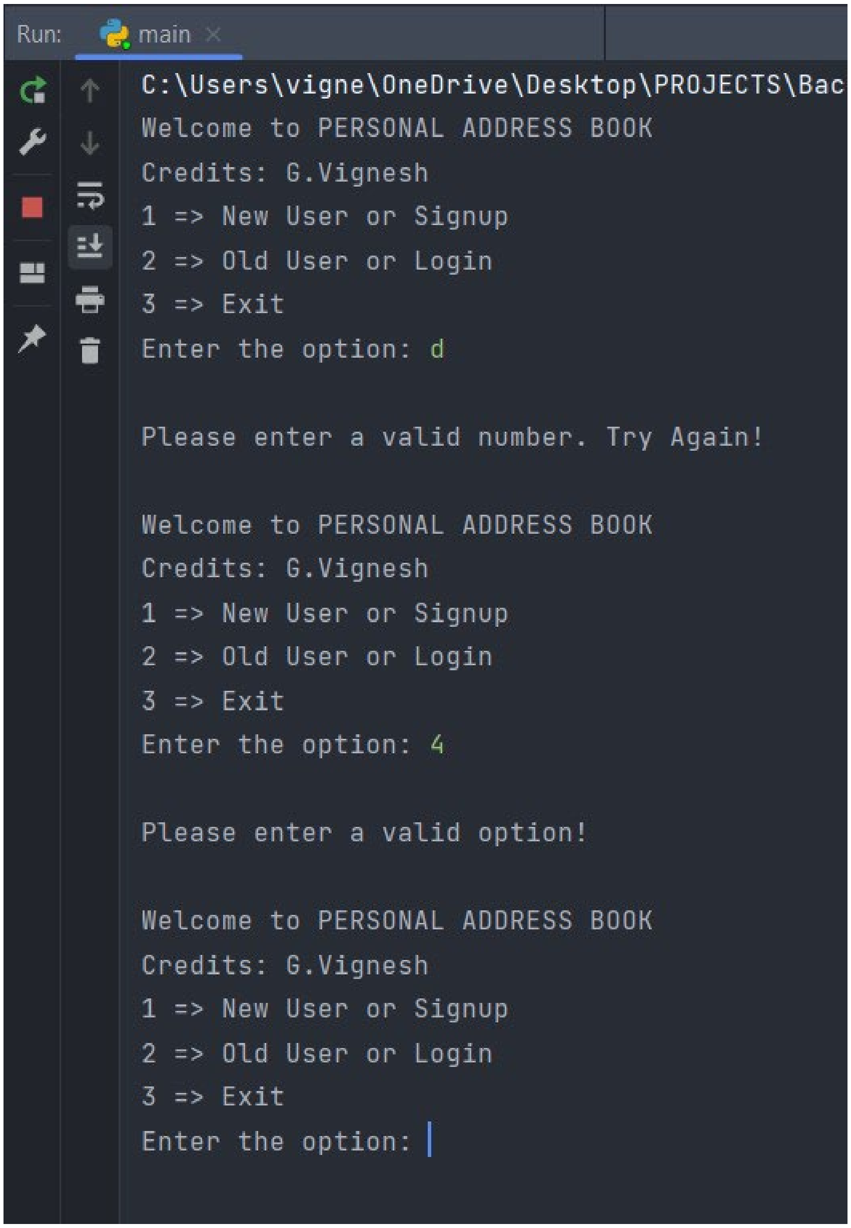
The app has multiple pages and it is built in such a way that it will exit the app only when the user enters an option to exit the app. Otherwise it will keeps redirecting to the various pages inside it.

The multiple pages are as follows:

- Homepage
- Sign Up Page
- Login Page
- Main menu Page
- Add entries Page
- View entries page
- Update Entries Page
- Delete Entries Page
- Exit Page

HOMEPAGE

The initial homepage will welcome the users and display the options like Signup, Login, and Exit app. If the user enters an invalid option, it will display and asks the user to enter a valid option or correct option. If the user enters the option1, it will redirect to a Sign-Up page. If the user enters the option2, it will redirect to a Login page. If the user enters the option3, it will exit the app.



```
C:\Users\vigne\OneDrive\Desktop\PROJECTS\Bac
Welcome to PERSONAL ADDRESS BOOK
Credits: G.Vignesh
1 => New User or Signup
2 => Old User or Login
3 => Exit
Enter the option: d

Please enter a valid number. Try Again!

Welcome to PERSONAL ADDRESS BOOK
Credits: G.Vignesh
1 => New User or Signup
2 => Old User or Login
3 => Exit
Enter the option: 4

Please enter a valid option!

Welcome to PERSONAL ADDRESS BOOK
Credits: G.Vignesh
1 => New User or Signup
2 => Old User or Login
3 => Exit
Enter the option: 
```

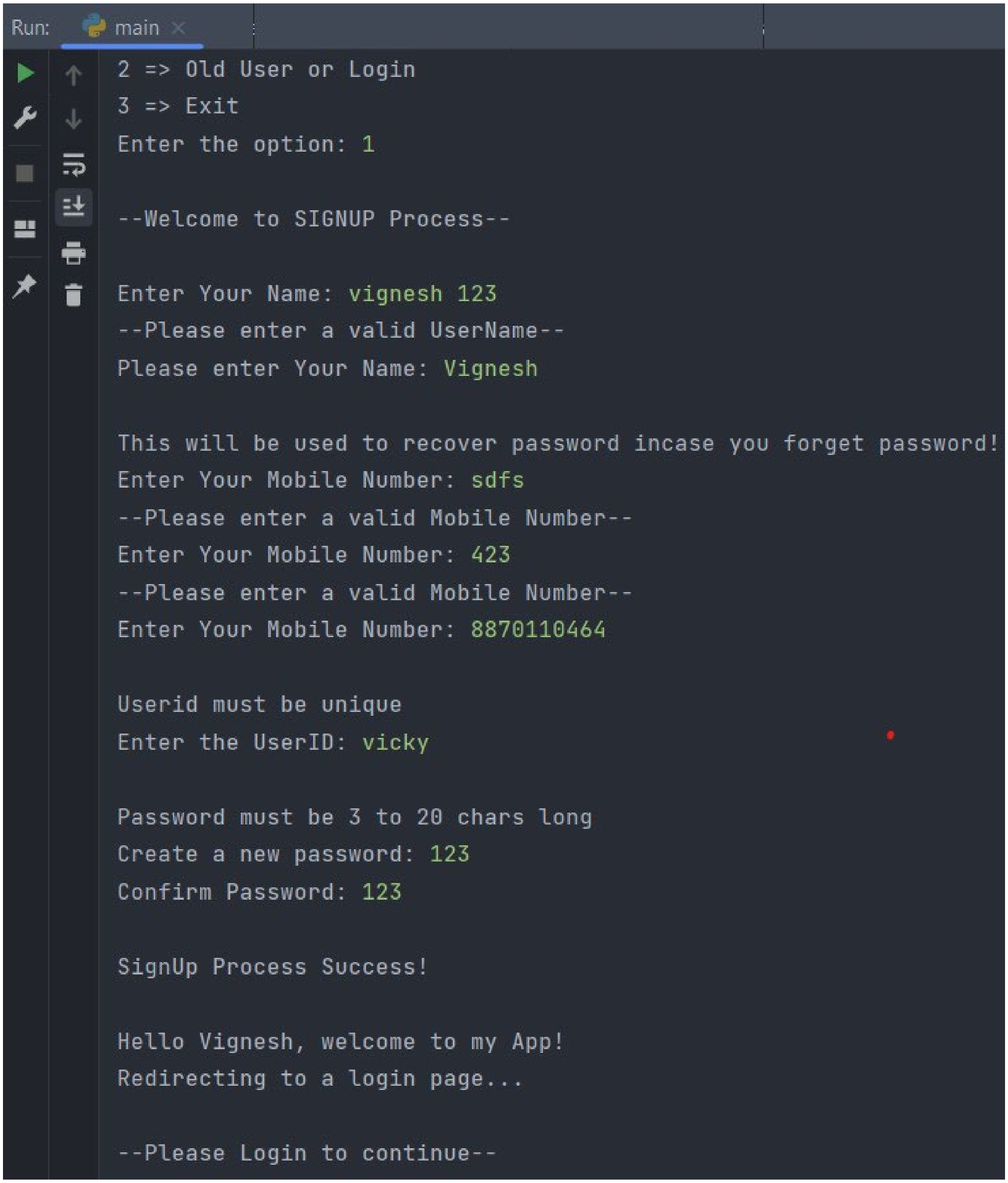
SIGNUP PAGE

When the user entered a correct option number 2, it will be redirected to the Signup page. In the Signup page, it asks for the user's name, user's mobile number, user's id, and user's password. It will keep asking for the user's name, user's mobile number, user's id, and user's password until the user enters the username in a correct format.

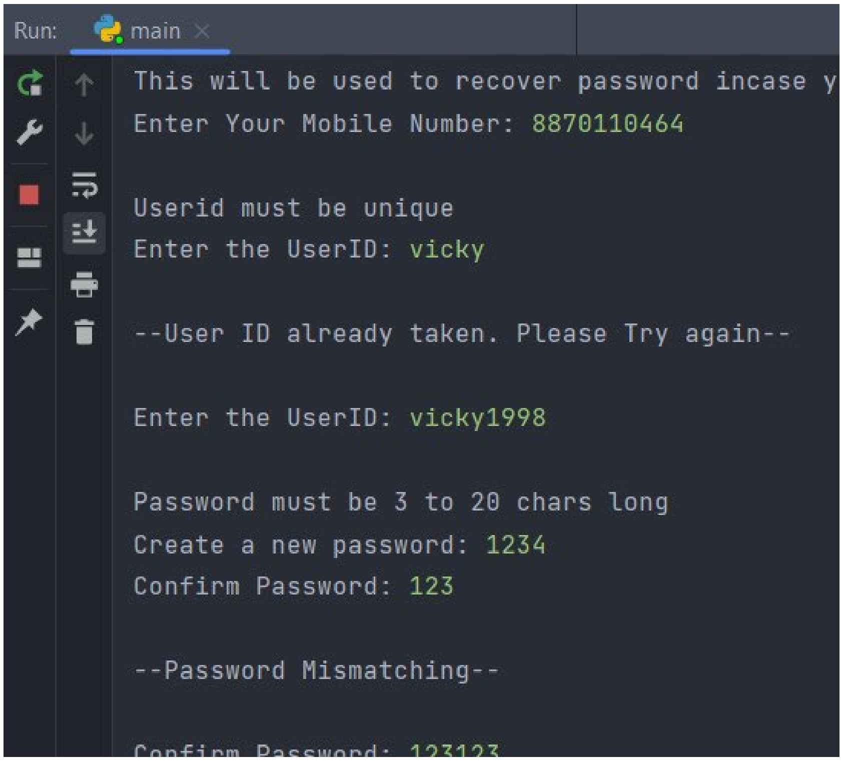
Correct Format:

- Username: Only Alphabets and between 3 to 30 chars long.
- User's mobile number: Only digits and minimum 10 chars long.
- User Id: Only Alphabets and between 3 to 20 chars long.
- User password: Only Alphabets and between 3 to 20 chars long.

When the user id is entered in correct format, it will check the availability of the UserId in the database. If the UserId is already in database, it asks the user to re-enter the UserId. After entering the unique UserId, it asks for the password and if the password is entered correctly it asks to re-enter for confirming password. If the user re-enters the password incorrectly, it will show that the password is mismatching and asks to re-enter the confirm password. After all the details are entered correctly, it will redirect to the login page.

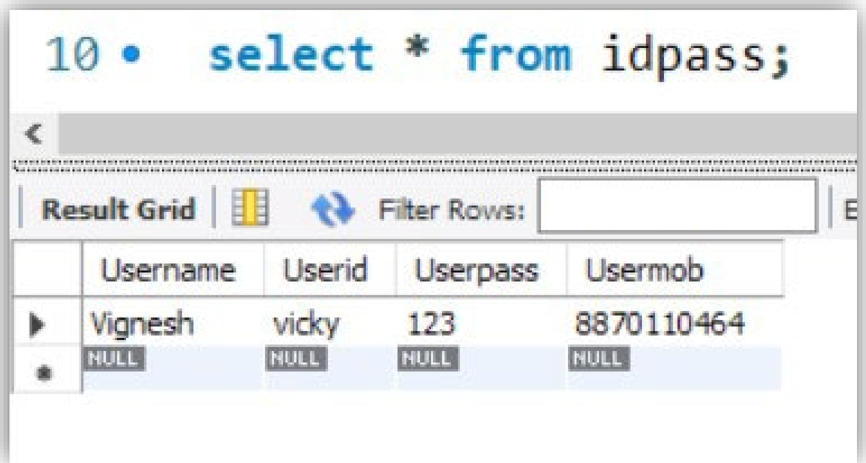


```
2 => Old User or Login
3 => Exit
Enter the option: 1

--Welcome to SIGNUP Process--

Enter Your Name: vignesh 123
--Please enter a valid UserName--
Please enter Your Name: Vignesh

This will be used to recover password incase you forget password!
Enter Your Mobile Number: sdfs
--Please enter a valid Mobile Number--
Enter Your Mobile Number: 423
--Please enter a valid Mobile Number--
Enter Your Mobile Number: 8870110464

Userid must be unique
Enter the UserID: vicky

Password must be 3 to 20 chars long
Create a new password: 123
Confirm Password: 123

SignUp Process Success!

Hello Vignesh, welcome to my App!
Redirecting to a login page...

--Please Login to continue--
```



```
This will be used to recover password incase y
Enter Your Mobile Number: 8870110464

Userid must be unique
Enter the UserID: vicky

--User ID already taken. Please Try again--

Enter the UserID: vicky1998

Password must be 3 to 20 chars long
Create a new password: 1234
Confirm Password: 123

--Password Mismatching--

Confirm Password: 123123
```

After successful creation of UserId, the username, UserId, Mobile Number and Password are saved into a separate table in a database.



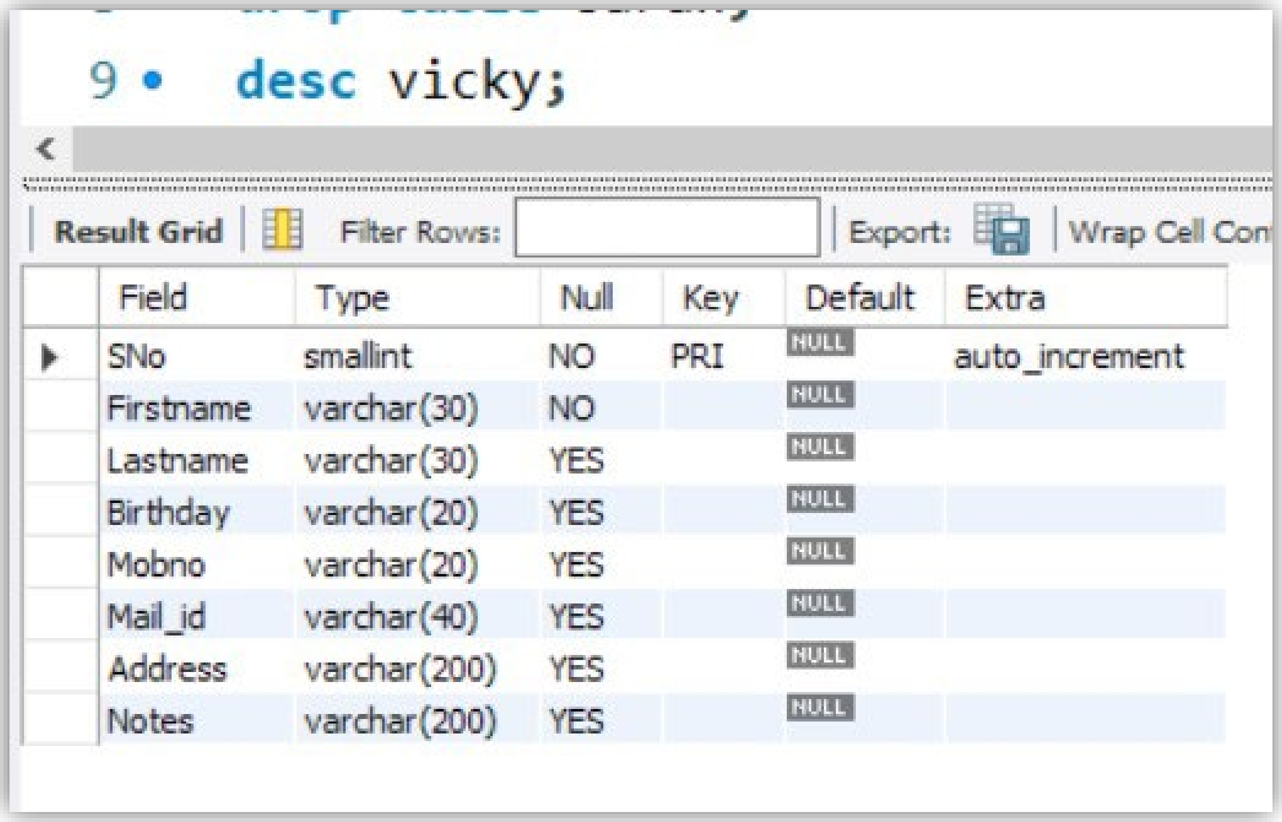
```
select * from idpass;
```

Username	Userid	Userpass	Usermob
Vignesh	vicky	123	8870110464
NULL	NULL	NULL	NULL

The separate table will also be created in the name of UserId where the entries are to be stored. That table is responsible for the creating, updating, viewing, and deleting the entries.

The table contains 7 columns as follows:

- First Name
- Last Name
- Birthday
- Mobile Number
- Mail id
- Address
- Personal Notes



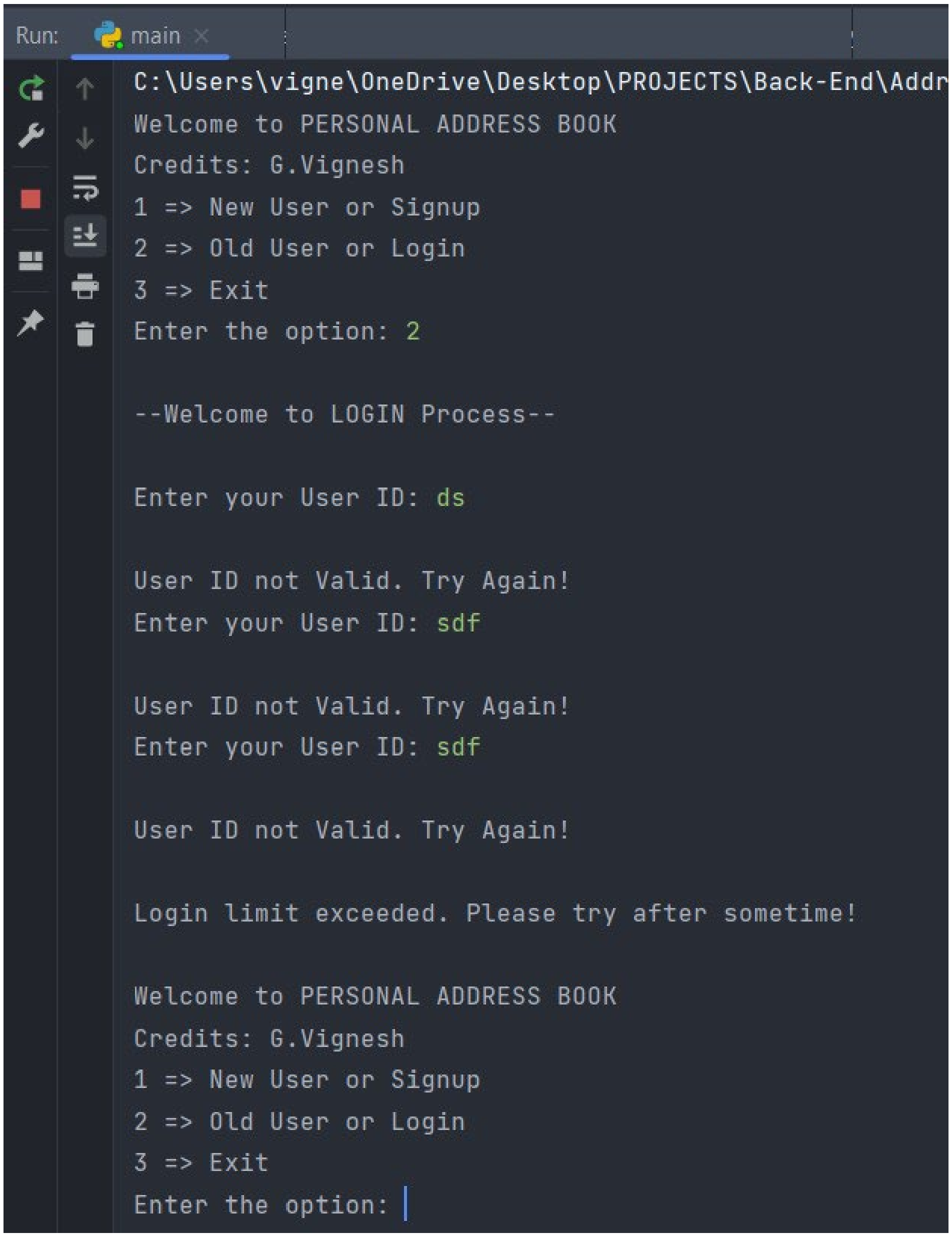
```
desc vicky;
```

Field	Type	Null	Key	Default	Extra
SNo	smallint	NO	PRI	NULL	auto_increment
Firstname	varchar(30)	NO		NULL	
Lastname	varchar(30)	YES		NULL	
Birthday	varchar(20)	YES		NULL	
Mobno	varchar(20)	YES		NULL	
Mail_id	varchar(40)	YES		NULL	
Address	varchar(200)	YES		NULL	
Notes	varchar(200)	YES		NULL	

LOGIN PAGE

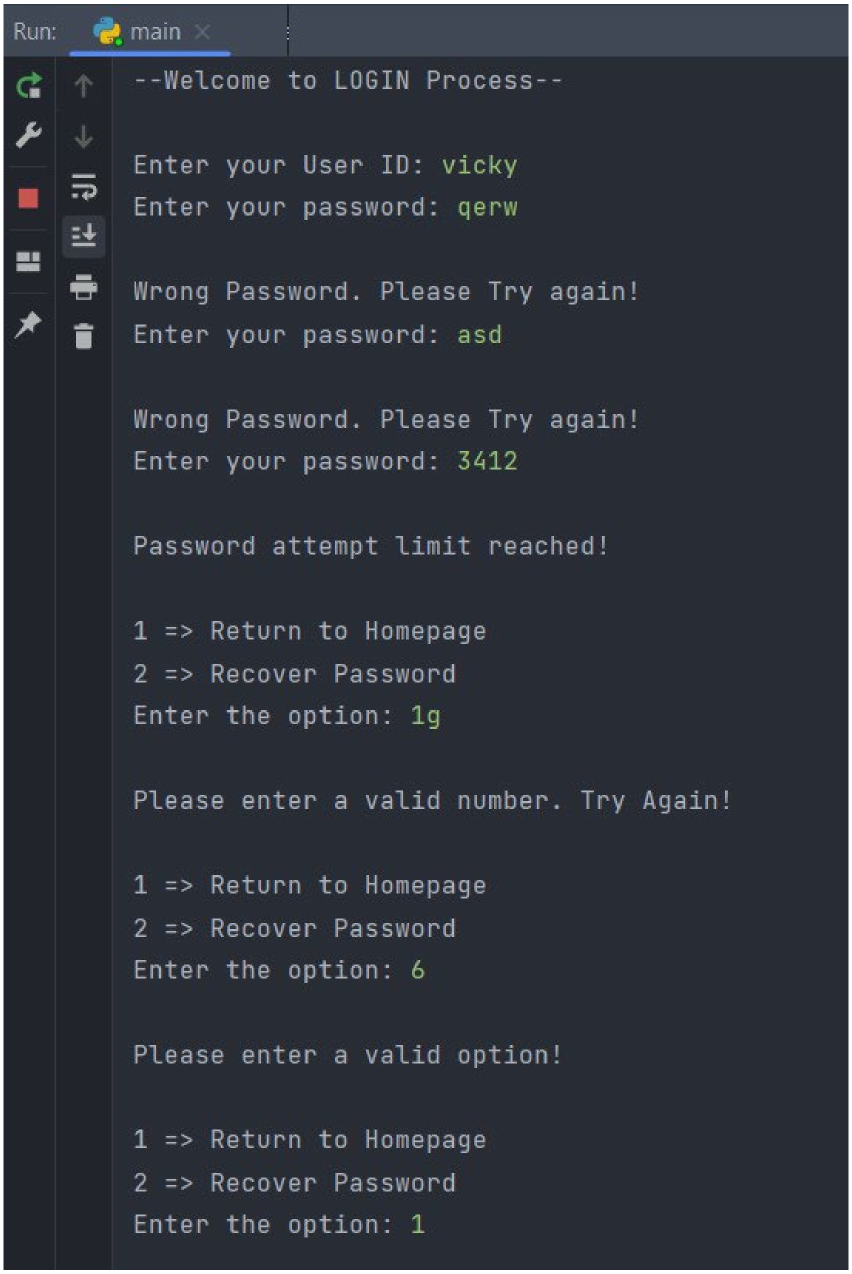
In the previous signup process, the username, UserId, mobile number and password are saved in a separate table in a database. In this login page, it initially asks for the UserId and checks in the database.

If the UserId is not found in the database, it shows the UserId is not found and if the user enters the unavailable UserId at 3 attempts wrongly, it redirects the user to the Homepage.



```
C:\Users\vigne\OneDrive\Desktop\PROJECTS\Back-End\Addr
Welcome to PERSONAL ADDRESS BOOK
Credits: G.Vignesh
1 => New User or Signup
2 => Old User or Login
3 => Exit
Enter the option: 2

--Welcome to LOGIN Process--

Enter your User ID: ds

User ID not Valid. Try Again!
Enter your User ID: sdf

User ID not Valid. Try Again!
Enter your User ID: sdf

User ID not Valid. Try Again!

Login limit exceeded. Please try after sometime!

Welcome to PERSONAL ADDRESS BOOK
Credits: G.Vignesh
1 => New User or Signup
2 => Old User or Login
3 => Exit
Enter the option:
```

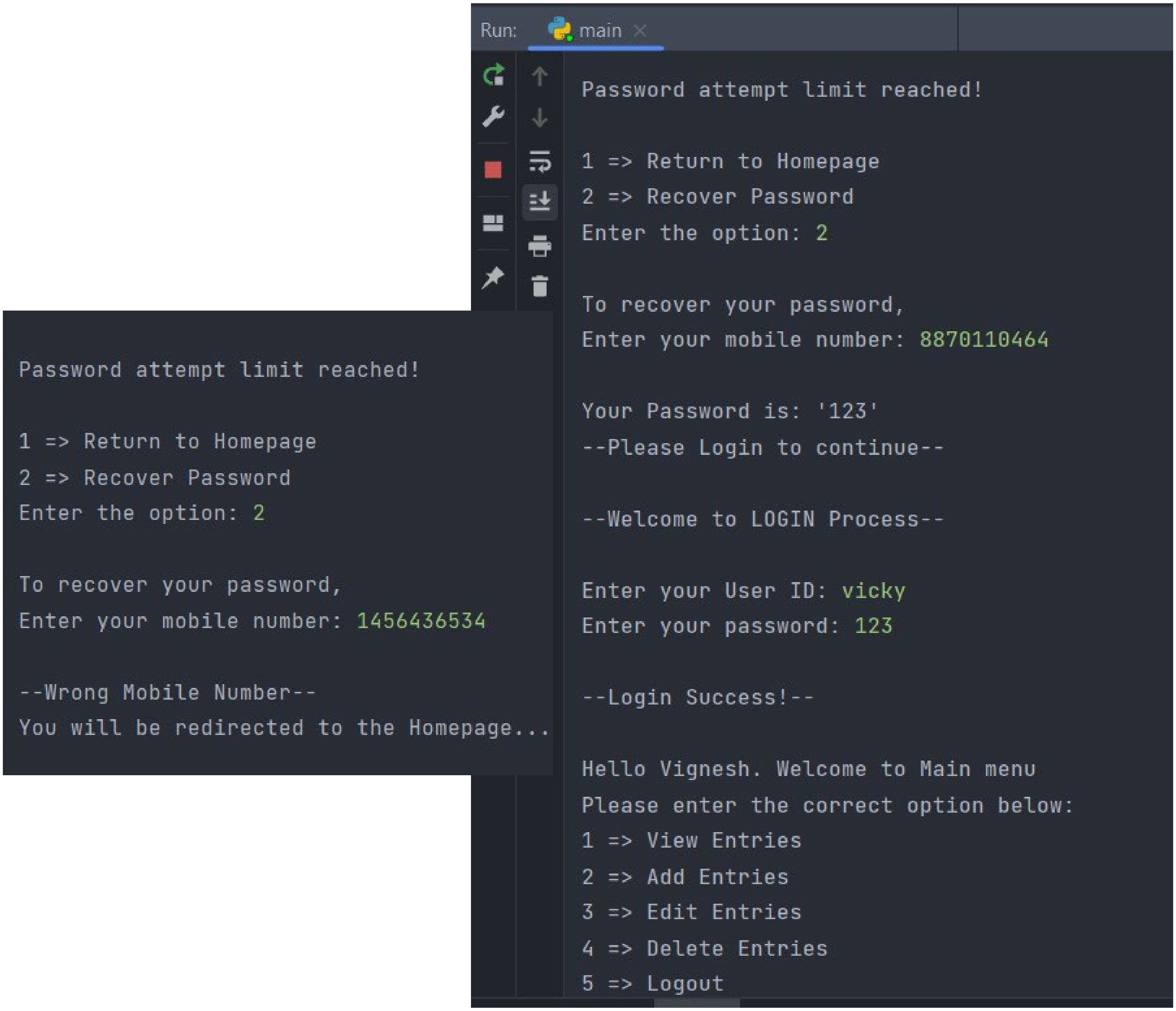
If the UserId is found in the database, then it will ask the user to enter the password. After the user entered password, it will check it in the database and if the entered password is wrong it shows wrong password and ask the user to re-enter the password. Whether the user enters the wrong password for 3 attempts, it will ask the user to enter the option. The options include 1 => Redirect to homepage and 2 => Forgot password. If the user enters the wrong option, it displays wrong option asks the user to re-enter the option and after entering the correct option1, it redirects to the homepage.



```
--Welcome to LOGIN Process--

Enter your User ID: vicky
Enter your password: qerw

Wrong Password. Please Try again!
Enter your password: asd

Wrong Password. Please Try again!
Enter your password: 3412

Password attempt limit reached!

1 => Return to Homepage
2 => Recover Password
Enter the option: 1g

Please enter a valid number. Try Again!

1 => Return to Homepage
2 => Recover Password
Enter the option: 6

Please enter a valid option!

1 => Return to Homepage
2 => Recover Password
Enter the option: 1
```

If the forgot password option2 is selected by the user, it asks the user to enter the mobile number. If the entered mobile number mismatches with the database or if it is in invalid, it redirects to the homepage. If the entered mobile number matches with the database, it shows the saved password from the database and redirects to login page.

After that user may enter correct UserId and password to login. Then the page will be redirected to the main menu where crud operations are done.

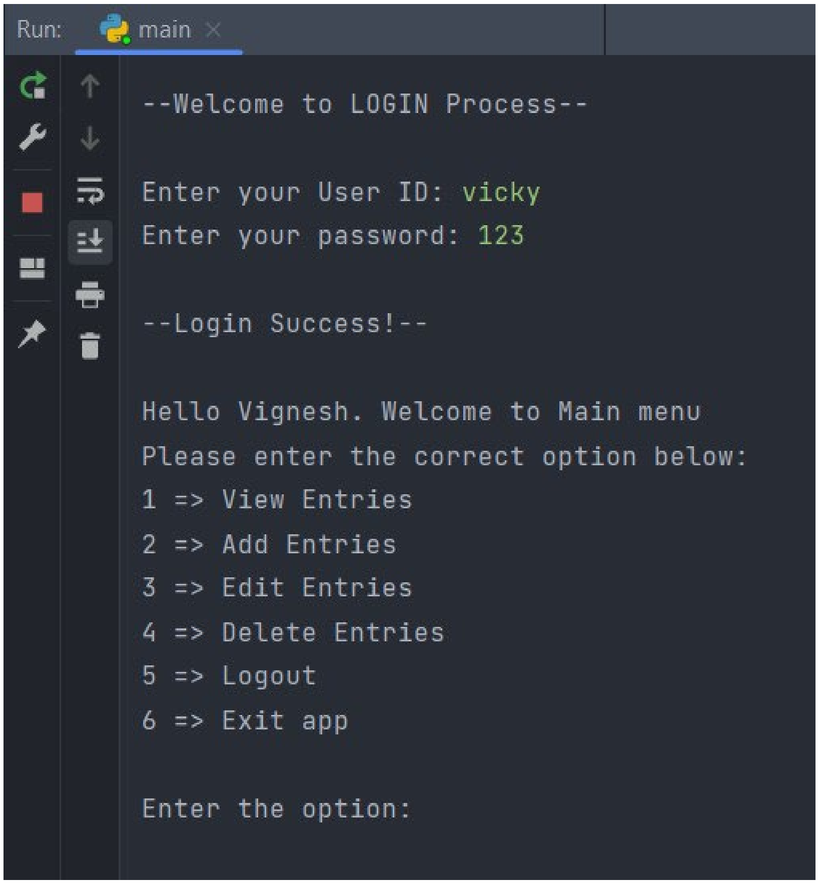


```
Password attempt limit reached!

1 => Return to Homepage
2 => Recover Password
Enter the option: 2

To recover your password,
Enter your mobile number: 1456436534

--Wrong Mobile Number--
You will be redirected to the Homepage...
```

```
Password attempt limit reached!

1 => Return to Homepage
2 => Recover Password
Enter the option: 2

To recover your password,
Enter your mobile number: 8870110464

Your Password is: '123'
--Please Login to continue--

--Welcome to LOGIN Process--

Enter your User ID: vicky
Enter your password: 123

--Login Success!--

Hello Vignesh. Welcome to Main menu
Please enter the correct option below:
1 => View Entries
2 => Add Entries
3 => Edit Entries
4 => Delete Entries
5 => Logout
```

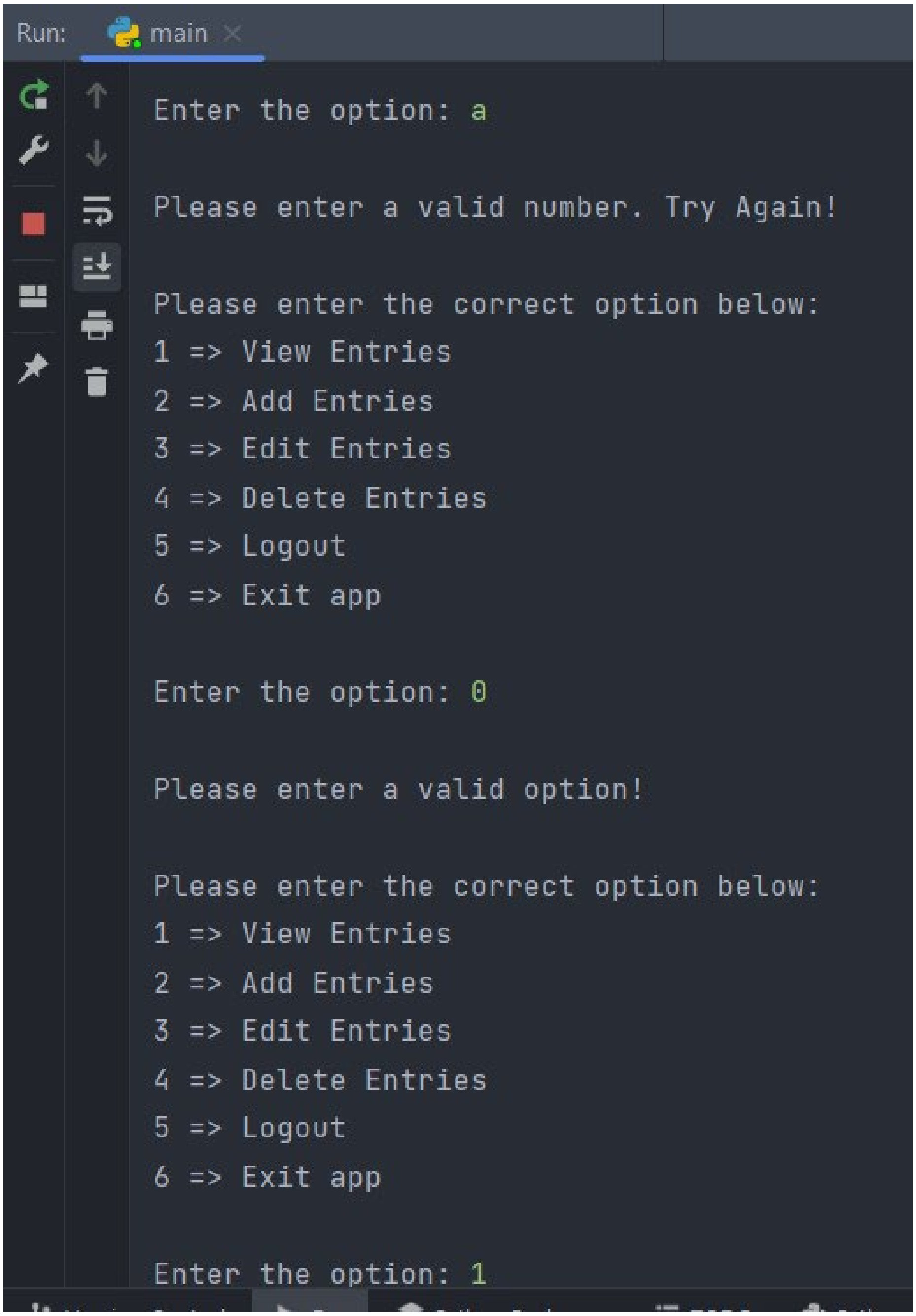
MAIN MENU PAGE

On entering the correct UserId and password, the user's name is obtained from the database and the main menu welcomes the user with the username. It will also display the list of options as mentioned in below image:



```
--Welcome to LOGIN Process--

Enter your User ID: vicky
Enter your password: 123

--Login Success!--

Hello Vignesh. Welcome to Main menu
Please enter the correct option below:
1 => View Entries
2 => Add Entries
3 => Edit Entries
4 => Delete Entries
5 => Logout
6 => Exit app

Enter the option:
```

It asks the user to enter the option from the above image as displayed and if the user enters an incorrect option or invalid option, it asks the user to re-enter the option again and again. The user can also logout from the main menu and go to the Homepage by entering the option no-5.

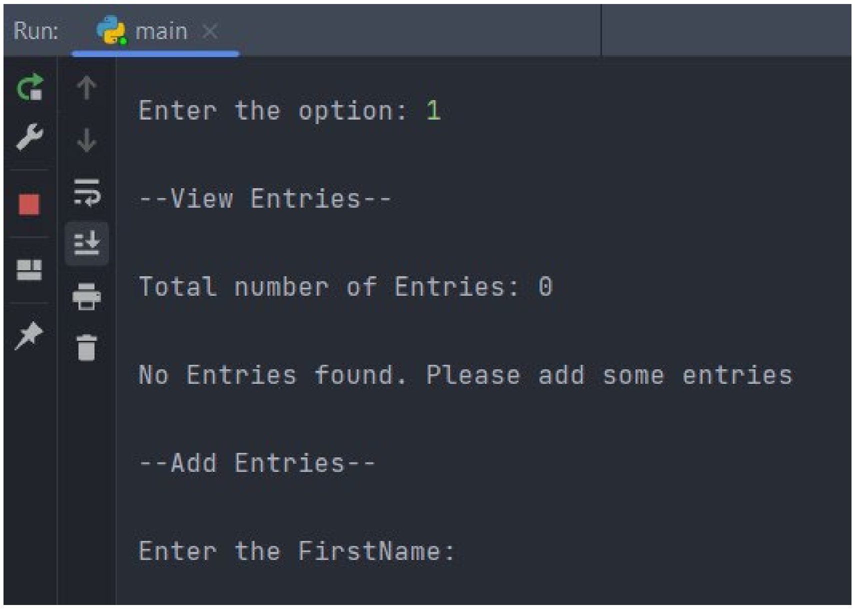


```
Enter the option: a

Please enter a valid number. Try Again!

Please enter the correct option below:
1 => View Entries
2 => Add Entries
3 => Edit Entries
4 => Delete Entries
5 => Logout
6 => Exit app

Enter the option: 0

Please enter a valid option!

Please enter the correct option below:
1 => View Entries
2 => Add Entries
3 => Edit Entries
4 => Delete Entries
5 => Logout
6 => Exit app

Enter the option: 1
```

It redirects to the respective pages from selected option only if the option is entered correctly.

VIEW ENTRIES PAGE

On entering the option no-1 it redirects to the View entries page which is the Read or View option from the CRUD app. Firstly it checks for the available data entries in the database and if there are no available data entries, it redirects to the Add entries page.

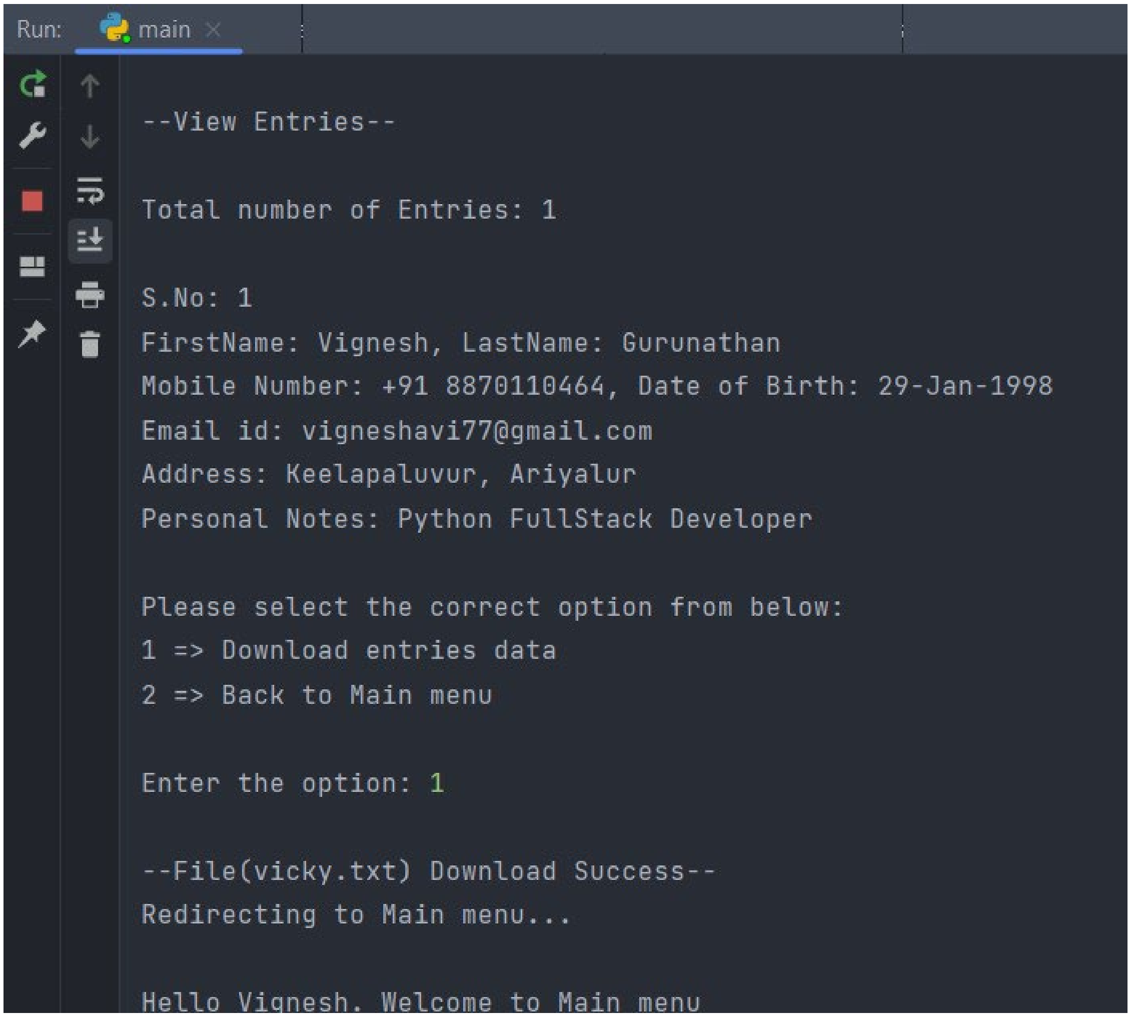


```
Enter the option: 1

--View Entries--

Total number of Entries: 0

No Entries found. Please add some entries

--Add Entries--

Enter the FirstName:
```

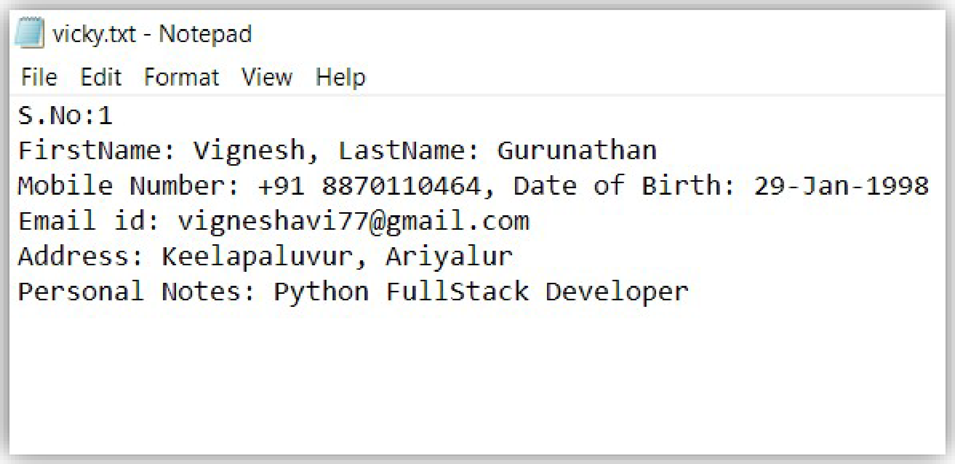
If there are any data entries in the database, it will show the data entries one by one and ask for two options as follows:

- Option no-1 => Download Entries data
- Option no-2 => Back to Main menu

If the user enters wrong option, it asks the user to re-enter the option and if the selected option is no:2 then it will redirect the user to the Main menu. If the entered option is 1 then the **.txt** file will be downloaded to the source folder in the name of the UserId and then it redirects the user to the Main menu.



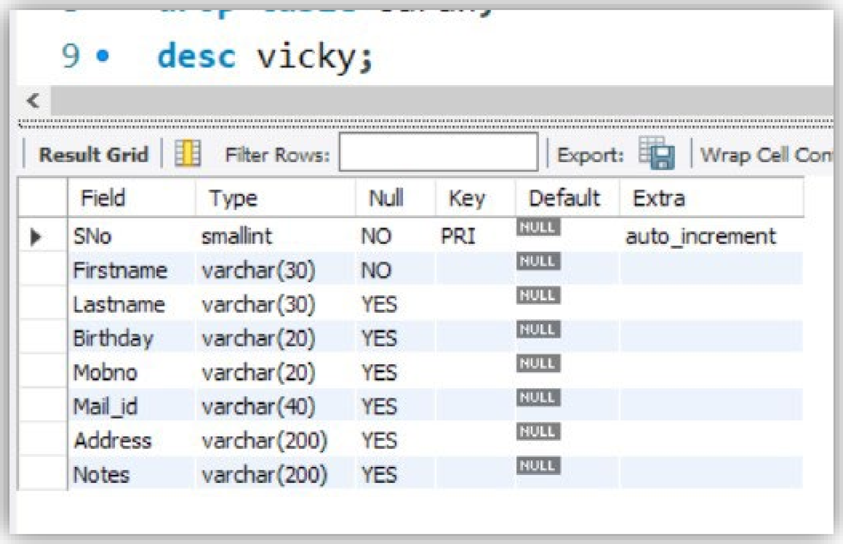
```
--View Entries--

Total number of Entries: 1

S.No: 1
FirstName: Vignesh, LastName: Gurunathan
Mobile Number: +91 8870110464, Date of Birth: 29-Jan-1998
Email id: vigneshavi77@gmail.com
Address: Keelapaluvur, Ariyalur
Personal Notes: Python FullStack Developer

Please select the correct option from below:
1 => Download entries data
2 => Back to Main menu

Enter the option: 1

--File(vicky.txt) Download Success--
Redirecting to Main menu...

Hello Vignesh. Welcome to Main menu
```



```
S.No:1
FirstName: Vignesh, LastName: Gurunathan
Mobile Number: +91 8870110464, Date of Birth: 29-Jan-1998
Email id: vigneshavi77@gmail.com
Address: Keelapaluvur, Ariyalur
Personal Notes: Python FullStack Developer
```

ADD ENTRIES PAGE

When the entered option is 2 in the main menu, the add entries page is displayed which is the Create option in the CRUD app. It is the section where the entries are created and stored into the databases. The required format of input are as follows from the MySQL workbench.

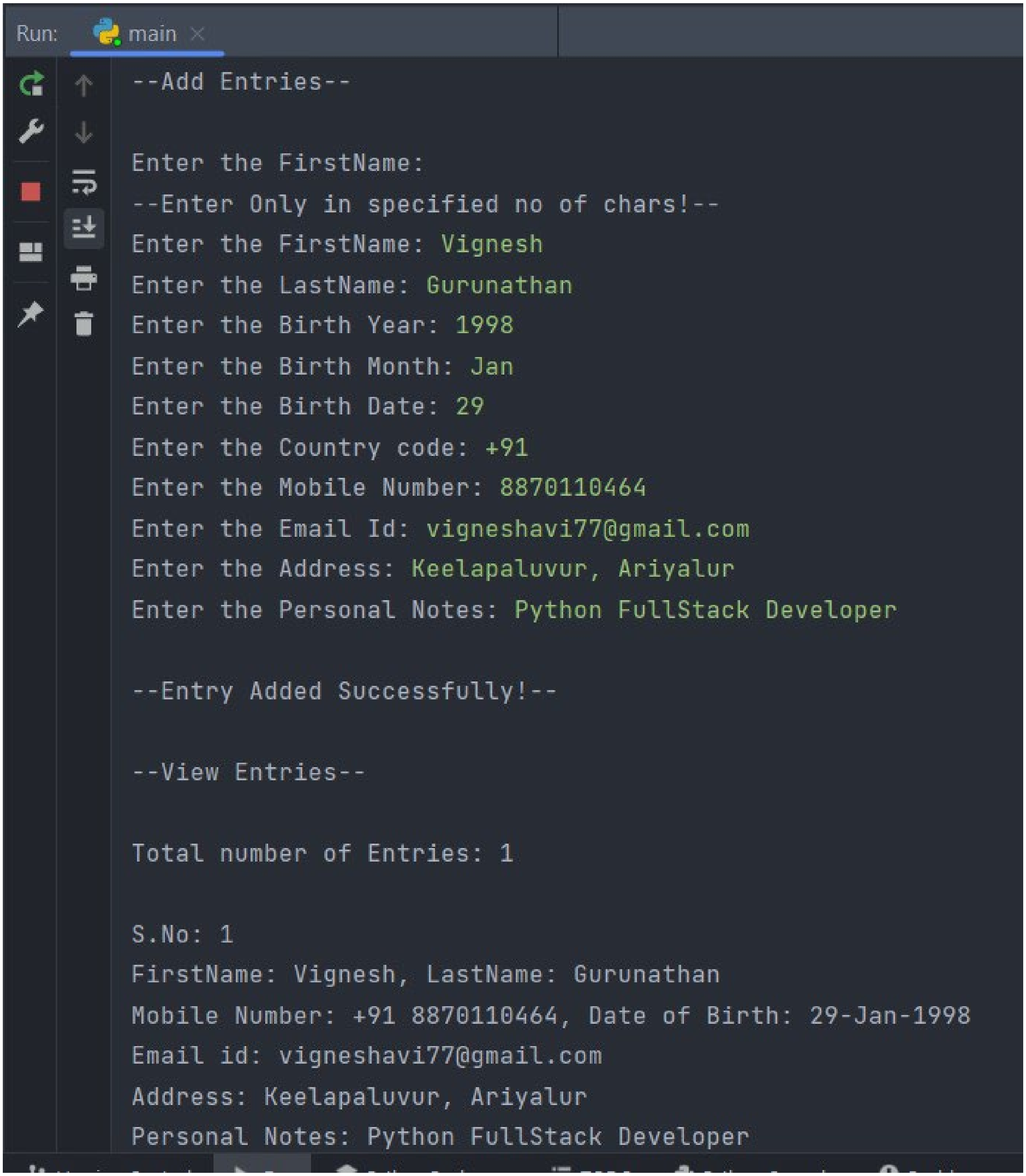


```
desc vicky;
```

Field	Type	Null	Key	Default	Extra
SNo	smallint	NO	PRI	NULL	auto_increment
Firstname	varchar(30)	NO		NULL	
Lastname	varchar(30)	YES		NULL	
Birthday	varchar(20)	YES		NULL	
Mobno	varchar(20)	YES		NULL	
Mail_id	varchar(40)	YES		NULL	
Address	varchar(200)	YES		NULL	
Notes	varchar(200)	YES		NULL	

The total of 8 columns out of which 7 columns are filled by the user's input and the first column (SNo) is an auto-increment column. Here it first asks the First Name which is mandatory because not null attribute is set in the MySQL database. Also, there are particular limitations indicated which are the no of characters.

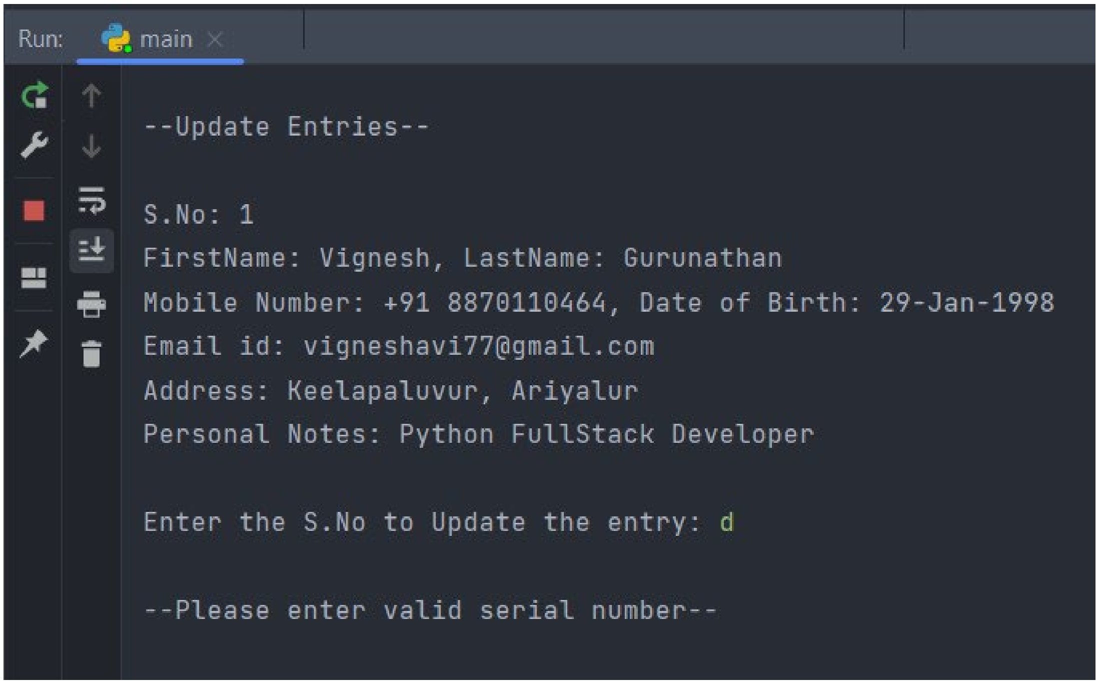
If the user enters the wrong no of characters, it will keep asking for the input values again and again. First name must not be null and the other values may or may not be null. After all the inputs are filled it will show the entries added.



```
--Add Entries--

Enter the FirstName:
--Enter Only in specified no of chars!--
Enter the FirstName: Vignesh
Enter the LastName: Gurunathan
Enter the Birth Year: 1998
Enter the Birth Month: Jan
Enter the Birth Date: 29
Enter the Country code: +91
Enter the Mobile Number: 8870110464
Enter the Email Id: vigneshavi77@gmail.com
Enter the Address: Keelapaluvur, Ariyalur
Enter the Personal Notes: Python FullStack Developer

--Entry Added Successfully!--

--View Entries--

Total number of Entries: 1

S.No: 1
FirstName: Vignesh, LastName: Gurunathan
Mobile Number: +91 8870110464, Date of Birth: 29-Jan-1998
Email id: vigneshavi77@gmail.com
Address: Keelapaluvur, Ariyalur
Personal Notes: Python FullStack Developer
```

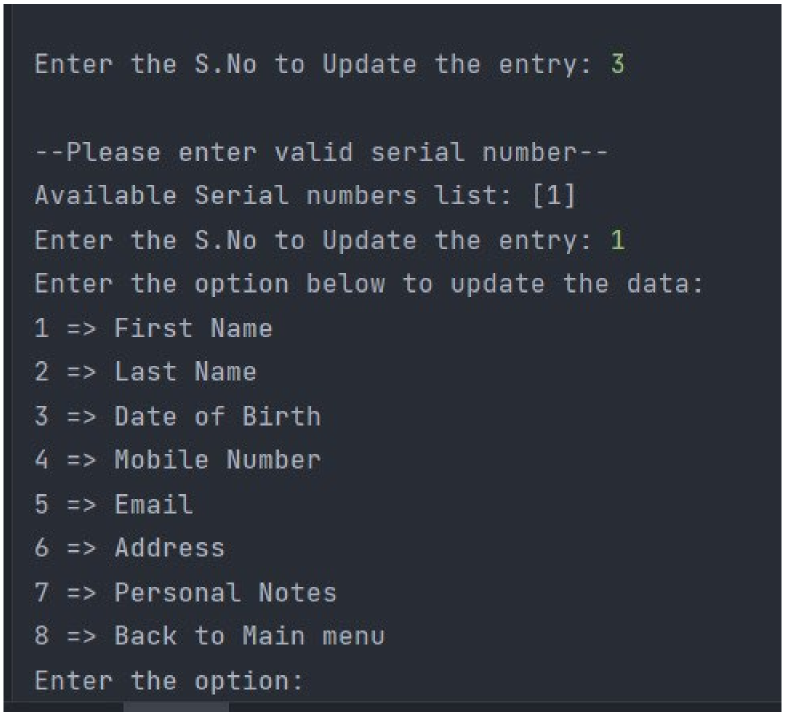
EDIT ENTRIES PAGE

When the entered option is 3 in the main menu, the edit entries page is displayed which is the Update option in the CRUD app. It is the section where the entries are edited and updated into the databases. Firstly, all the data entries available in the databases are displayed with the SNo. After that it asks the SNo to update the data entry and if the user entered the incorrect option or invalid option, it asks the user to re-enter the correct option.



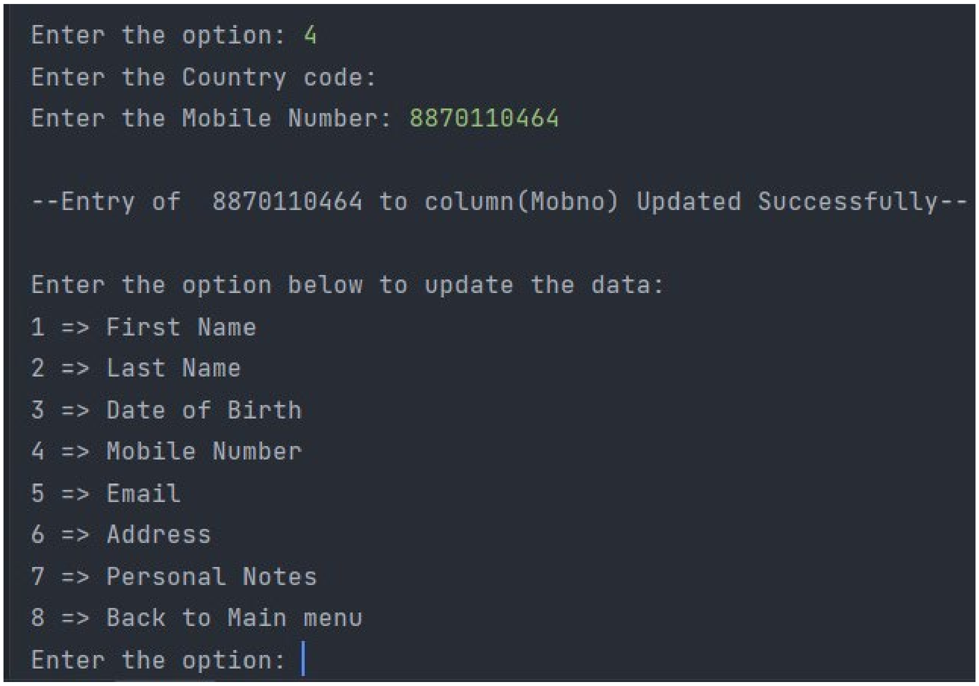
```
--Update Entries--

S.No: 1
FirstName: Vignesh, LastName: Gurunathan
Mobile Number: +91 8870110464, Date of Birth: 29-Jan-1998
Email id: vigneshavi77@gmail.com
Address: Keelapaluvur, Ariyalur
Personal Notes: Python FullStack Developer

Enter the S.No to Update the entry: d

--Please enter valid serial number--
```

It also shows the list of available serial numbers in the database if the incorrect SNo is entered. After entering the correct serial number from the list, it shows the list of options to select from which entry is to be edited or updated. The user has to enter the correct option to update the entries.



```
Enter the S.No to Update the entry: 3

--Please enter valid serial number--
Available Serial numbers list: [1]
Enter the S.No to Update the entry: 1
Enter the option below to update the data:
1 => First Name
2 => Last Name
3 => Date of Birth
4 => Mobile Number
5 => Email
6 => Address
7 => Personal Notes
8 => Back to Main menu
Enter the option:
```

After selecting the correct option to update the entry, the entry will be updated and displays that the entry has been updated and it also asks the next option for the entry to be updated. After updating the required entries from the SNo, we can back to Main menu by selecting option 8.



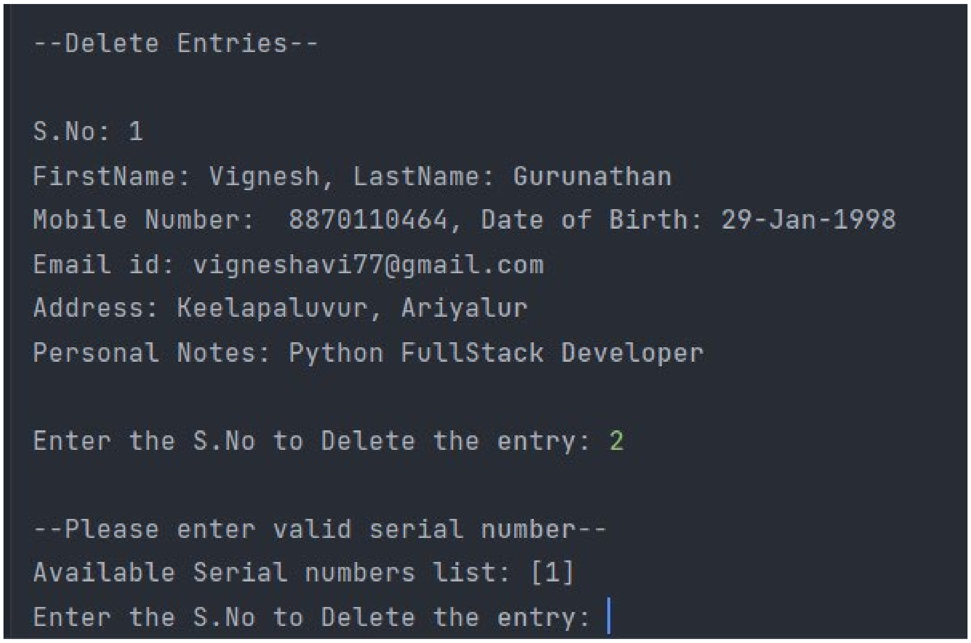
```
Enter the option: 4
Enter the Country code:
Enter the Mobile Number: 8870110464

--Entry of  8870110464 to column(Mobno) Updated Successfully--

Enter the option below to update the data:
1 => First Name
2 => Last Name
3 => Date of Birth
4 => Mobile Number
5 => Email
6 => Address
7 => Personal Notes
8 => Back to Main menu
Enter the option:
```

DELETE ENTRIES PAGE

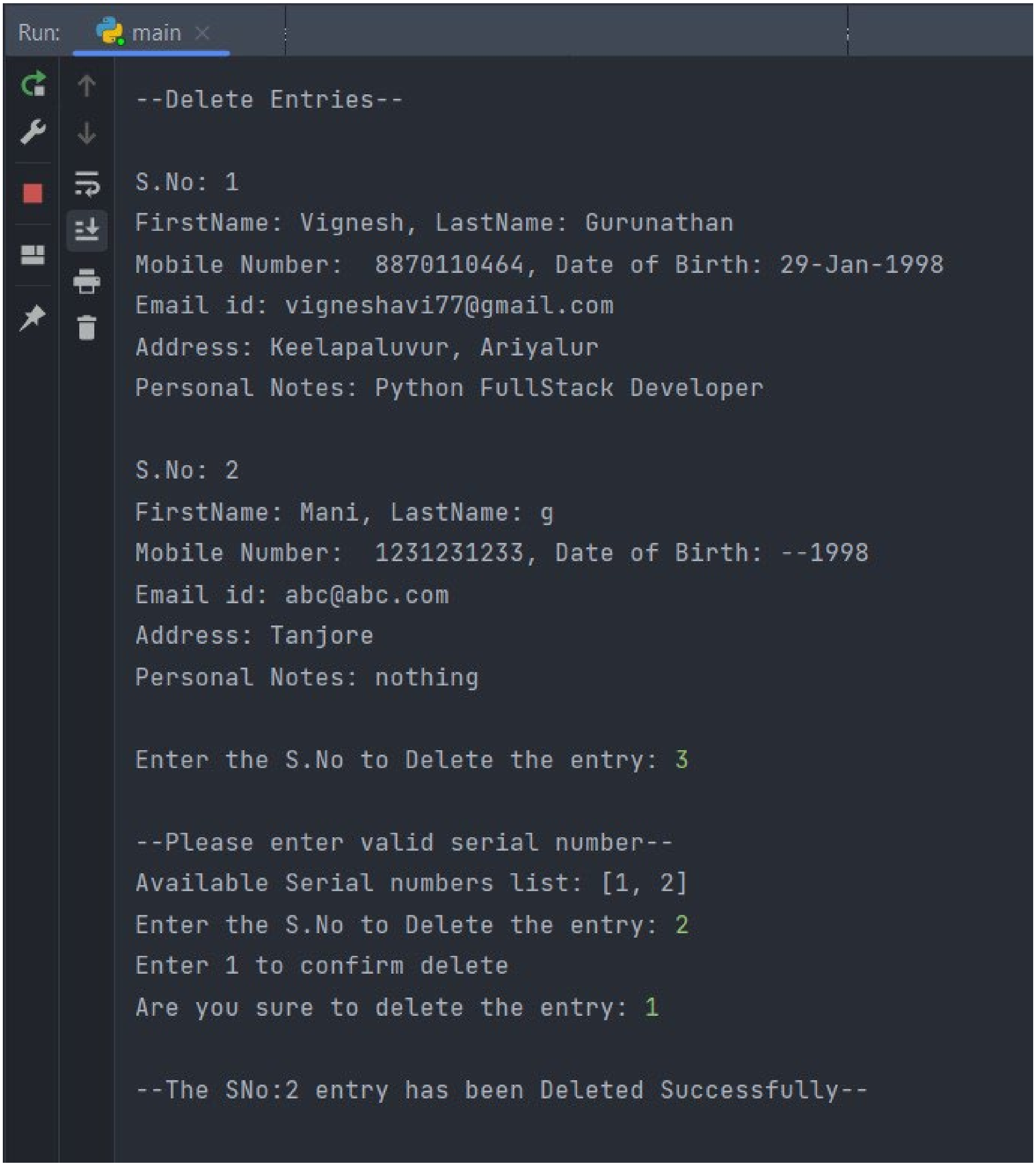
When the entered option is 4 in the main menu, the delete entries page will be displayed which is the Delete option in the CRUD app. It is the section where the entries are deleted from the databases.

Firstly, all the data entries available in the databases are displayed with the SNo. After that it asks the SNo to delete the data entry from the database and if the user entered the incorrect option or invalid option, it asks the user to re-enter the correct option.

On entering the correct SNo it asks again to confirm the delete operation and if the user enters the correct option, the data entry from the database will be deleted and it shows the SNo with entry deleted. Then it Redirects to the Main menu. If the user entered incorrect option it displays the list of available SNo from the database and asks the user to re-enter the SNo.



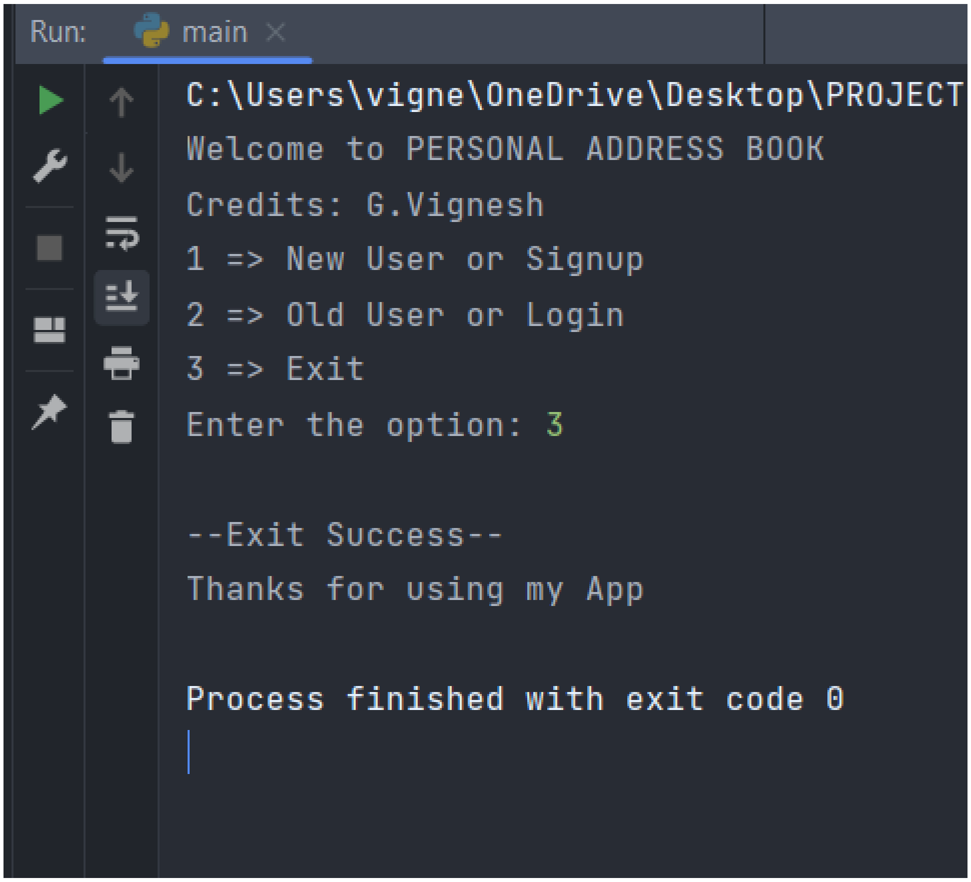
```
--Delete Entries--

S.No: 1
FirstName: Vignesh, LastName: Gurunathan
Mobile Number:  8870110464, Date of Birth: 29-Jan-1998
Email id: vigneshavi77@gmail.com
Address: Keelapaluvur, Ariyalur
Personal Notes: Python FullStack Developer

Enter the S.No to Delete the entry: 2

--Please enter valid serial number--
Available Serial numbers list: [1]
Enter the S.No to Delete the entry: 
```



```
--Delete Entries--

S.No: 1
FirstName: Vignesh, LastName: Gurunathan
Mobile Number:  8870110464, Date of Birth: 29-Jan-1998
Email id: vigneshavi77@gmail.com
Address: Keelapaluvur, Ariyalur
Personal Notes: Python FullStack Developer

S.No: 2
FirstName: Mani, LastName: g
Mobile Number:  1231231233, Date of Birth: --1998
Email id: abc@abc.com
Address: Tanjore
Personal Notes: nothing

Enter the S.No to Delete the entry: 3

--Please enter valid serial number--
Available Serial numbers list: [1, 2]
Enter the S.No to Delete the entry: 2
Enter 1 to confirm delete
Are you sure to delete the entry: 1

--The SNo:2 entry has been Deleted Successfully--
```

EXIT PAGE

When the entered option in the main menu is 5 it redirects to the homepage of the app and if the entered option is 6, the display message will be shown as “Exit success and Thanks for using my app”. It will exit the interpreter with exit code of “Process finished with exit code 0”.



```
C:\Users\vigne\OneDrive\Desktop\PROJECT
Welcome to PERSONAL ADDRESS BOOK
Credits: G.Vignesh
1 => New User or Signup
2 => Old User or Login
3 => Exit
Enter the option: 3

--Exit Success--
Thanks for using my App

Process finished with exit code 0
```

CODING DISCUSSIONS

- Initially we run the *main.py* file where it contains the imported file named *hompepage.py* which has the function *welcome( )*. This function welcomes the user and asks the user to enter the option for *Signup, Login or exit the app.* It also defines the exceptional handling to check the entered option. Welcome( ) function is called frequently again and again when the exception is there or user enters the wrong option by using *Recursion* function.

- If the user enters the correct option, it calls the respective functions by using *match case.* If option 1 is selected it calls the function from the module named *signup.py* which has the function named *newUser( )*. This function is used to get the username, UserId, Mobile Number, and User password. It also checks the users input by match function in the *Regular Expression* module and allows the user only when it matches with the correct format using while loop.

- When all the entries right it stores in variables and supply it in a function belongs to the *Class* named *create_new_user(userid)* which is inherited by the super class named *userid( )*. The super class userid( ) has the constructor which defines the connection between the MySQL database and Python Interface using **pymysql** module.

- The constructor in the *inherited class* *creates_new_user(userid)* is supplied with function arguments of user's input and the function *create_user( )* is used to store the username, UserId, Mobile Number, and User password in the table named idpass in MySQL database and also it creates the separate table in the name of UserId to store the user's Data Entries.

- After the successful creation of the UserId it calls the function *oldUser( )* from the module named *login.py* which welcomes the user and asks to enter UserId. When the user enters an UserId it checks in the database for the UserId in table named *idpass* by using the function *checkRegid( )* which returns the list of available UserId from the database using the function *get_idlist( )* from the *module* *mysql_idpass.py.*

- If the user id is in the returned id list, then it asks to input password and it *checks* for the password matching using the function *get_password(uid)* in the *super class* *UserId( )*. The function *checkUpass(uid,upass)* checks the password and returns the *Boolean value*. After three wrong attempts of password, it asks the user's mobile number to recover password and it supplies the UserId and mobile number in the function *check_umob(uid,upass)* from the module *mysql_idpass.py* and returns the Boolean value. If the Boolean value is True, it shows the password from the database else it will call the function to homepage.

- After successful login, the function *menu_page(uid)* with UserId as an argument is called where it asks the user to enter options in menu. It is operated using a *match case* with the respective functions called.

- If the entered option is **1**, it redirects to function *view_entries(uid)* in the *crud_r.py* module. If the entered option is **2**, it redirects to function *add_entries(uid)* in the *crud_c.py* module. If the entered option is **3**, it redirects to function *edit_entries(uid)* in the *crud_u.py* module. If the entered option is **4**, it redirects to function *delete_entries(uid)* in the *crud_d.py* module. If the entered option is **5**, it redirects to function *welcome( )* in the *homepage.py* module. If the entered option is **6**, it redirects to function *exitApp( )* in the *quit.py* module.

- The *Class users_table( )* is created in the *mysql_entries.py* module with *constructor* to connect to the database using *pymysql* module and various functions in it to add, view, edit and delete the entries of the database.

- The function *view_entries* in *crud_r.py* module checks for the entries by using the function *view_table(uid)* in the class *users_table( )*. It lists all the entries in the database. If there are no entries in the database, it shows no entries found and calls the function *add_entries(uid)* in the *crud_c.py* module.

- If the entries are found and after displaying the entries, it asks the user whether he wants to download the entries. If the user enters a correct option to download entries, it matches with case and file handling sequences will be generated to save the file to the directory with name of the *UserId .txt* format.

- The function *add_entries( )* in *crud_c.py* module asks and store all the details to be entered in separate variables and supply as the arguments in the function *create_entry( )* from the class *users_table( )* in the *mysql_entries.py* module. This function will insert the respective values to the respective columns and store it in a UserId's table in the database.

- The function *edit_entries(uid)* in *crud_u.py* module initially shows all the data entries from the database with *SNo* and asks the user's input for SNo to edit the entry. When correct SNo is entered, it calls the function *update_entry(uid, sno)* which supplies *user's id and SNo* as arguments and it asks user's input to which entry is to be updated.

- On entering the valid option, it matches with the appropriate cases and calls the function *update_entry(*args)* in the *class users_table( )* from the *mysql_entries.py* module with the arguments of *UserId, Option to update, Value to be updated, and the SNo*. Then the values will be applied successfully updated in the UserId's table of the database.

- Similarly, the function *delete_entries( )* in *crud_d.py* module displays all the data entries from the database with SNo and asks the user's input for SNo to delete the entry. When correct SNo is entered, it calls the function *del_entry(uid, sno)* which supplies user's id and SNo as arguments and it asks user's input to which entry is to be deleted.

- On entering the valid option, it matches with the appropriate cases and calls the function *delete_entry(uid, SNo)* in the class *users_table( )* from the *mysql_entries.py* module with the arguments of *UserId, and the SNo to be deleted*. Then the entered SNO entries will be deleted from the UserId's table in the database.

- The function *exitApp( )* from the *quit.py* module will finish all the process and exit the python interface with the *exit code 0* and no errors will be taken place.

CONCLUSION

Hereby we can conclude that the CRUD app has been created and executed successfully without any errors. We can use this app to store and access all the options in the app by using CRUD operations. From the foundation of account creation process and Login process to all the CRUD operations of Create, Read, Update and Delete operations has been performed fluently without any errors. This app is built in a way without any complications that the user has to just enter the options and values accordingly. All the CRUD operations are done with the Python interface as a Front-end and the MySQL database as a Back-end tools.