|  |  |  |
| --- | --- | --- |
| Quiz **Book** | |  |
|  | | |
| Done byRanjith R09/12/2022 |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table of Contents:   |  |  | | --- | --- | | Introduction to Quiz Book | 03 | | Security | 04 | | Tables for the application | 05 | | Admin tab | 05 | | User tab | 06 | | Quiz | 07 | | Python Codes, Packages and Modules | 08 | | Quiz Book Application Structure | 09 | | Login Page | 10 | | Signup page | 11 | | Admin homepage | 12 | | User operations | 12 | | Quiz Operations | 16 | | User homepage | 20 | | Profile | 20 | | Take Quiz | 21 | | Instructions | 23 | | Conclusion | 24 | | | |
| **Introduction:**  Quiz Book application is an application where we can interact with question and select answers to test our skill and learn new skills as well.  In this Project, we have created a new quiz book application using Python and MySQL to do the same but in a computer terminal..  We used MySQL database to create tables, and store all the related data of this quiz book application.  And we used Python for all the calculation, interacting with database, and interacting with the application. | | |
|  | At the end of the day, the goals are simple: Safety and Security. -Jodi Rell |

**Security**

When coming to securing the application, the best way to do this is making passwords. To make it more secure, separating the users and admin and other type of users. We have created multiple tables to separate admins and users to make sure there won’t be any password leak, or insecurity.

**Data base:**

We have created and used our own database called “quiz\_book” to implement all the tables and data in it.

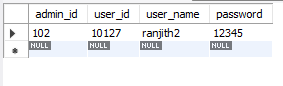
**Tables for the application:**

We have created multiple tables to store data as per the application need. Let’s see the tables an their characters.

Tables:

* + - admin\_tab
    - user\_tab
    - quiz

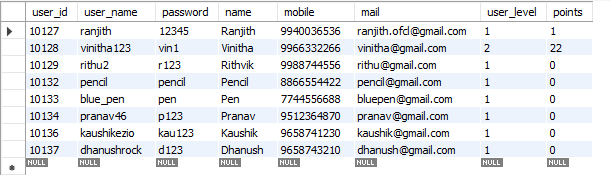
**admin\_tab:**

****

|  |  |
| --- | --- |
| admin\_id | int |
| **user\_id** | int AI PK |
| user\_name | varchar(45) |
| password | varchar(45) |

Admin\_tab is the table which has all the information of administrator. As we set the user\_id as primary key, there will be a new user created in user\_tab whenever there is a new admin created. Admin can also be part of taking quiz and using the application while logging in as user. Admin\_tab and user\_tab shares the same password for admin.

**user\_tab:**



Columns:

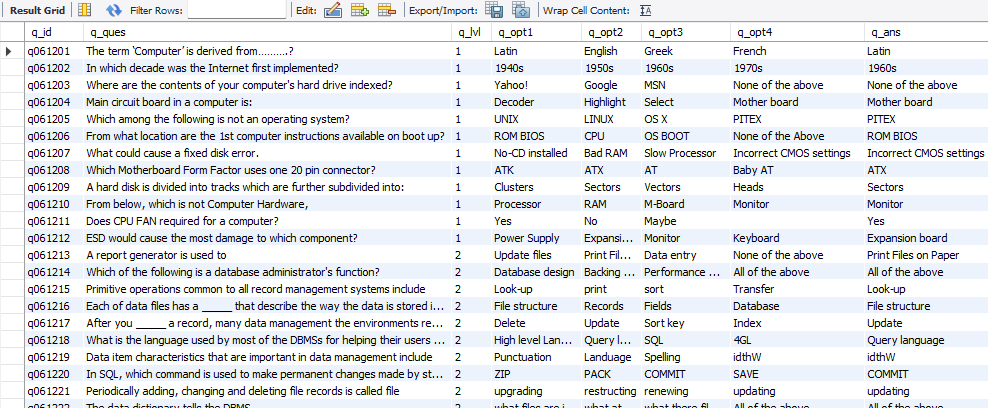
|  |  |
| --- | --- |
| **user\_id** | int AI PK |
| user\_name | varchar(45) |
| password | varchar(45) |
| name | varchar(45) |
| mobile | varchar(45) |
| mail | varchar(45) |
| user\_level | int |
| points | int |

User\_tab is the table is hold all the user information. As user-id primary key, we can also interact with other tables.

Here we can see user\_level and points columns as int. There are the columns which makes this application more interesting.

Users can take more quizzes as possible to increase their level. Their points will be used to evaluate their level. More level comes with more difficult quizzes.

**Quiz:**

****

Columns:

|  |  |
| --- | --- |
| **q\_id** | varchar(20) PK |
| q\_ques | varchar(2000) |
| q\_lvl | int |
| q\_opt1 | varchar(1000) |
| q\_opt2 | varchar(1000) |
| q\_opt3 | varchar(1000) |
| q\_opt4 | varchar(1000) |
| q\_ans | varchar(1000) |

Quiz table is the table which holds all the quiz questions, answers, quiz level. Here we can see q\_lvl column. This is the difficulty level of quiz. We will see more about this lvl column and use of it in the Quiz features.

**Python Codes, Packages and Modules:**

**Python Code:**

Python is a programming language that enables users to easily write and read code. It is easy to learn for beginners and has many features that make it useful for experienced programmers.

**Python Package:**

A Python module is a file that has a .py extension, and it is a program that is written in the Python programming language. The module can be run by calling the Python interpreter with the -m option, and it will be executed as if it were a script.

**Python Module:**

A Python package is a collection of modules that are bundled together and distributed as a single unit. Packages can contain code, data, and documentation.

**In our application:**

In this application, we have used modules, packages to make the application more intractable, easy to debug, easy to test. Every feature of this application comes in a new package/module as what is best location for them.

Here is the diagram to explain, how this codes, modules, packages are stored and used.

By making the login and signup page as main code, we are improving the security as user can only be as user, and admin can be user and admin as well. Also this helps us to dive into the application directly without any unnecessary steps.

**Quiz Book Application Structure:**

Login page

User home

Admin home

Features

User home

User operations

Quiz operations

Profile

Add Quiz

Add user

Take Quiz

View Quiz

View Users

Instructions

Delete Quiz

Modify Quiz

Search Quiz

Delete Users

Modify Users

Search Users

**Login Page: (login\_page.py)**

Login page is the beginning of this application. Here we will have four options as follows,

\* Login as Admin

\* Login as User

\* Signup

\* Exit

1. Login as Admin option will be used for admin login. Here admin\_tab table is used to retrieve admin ID, and password. There are restrictions as password incoorect, no admin found (If the admin ID is incorrect). Let’s see the code and query we used to make this page,

"select password from admin\_tab where admin\_id = %s"

If the query retrieves no admin, we can be sure there is no admin available with the entered admin id. It prompts “No Admin Found” error.

Once the admin and password is retrieved, if the password mismatch with the entered password and retrieved password, then it prompts “Password incorrect” error.

If both admin id and password is correct, verification will be successful and the page directs to “admin\_home.py” module where admin can access all admin features.

2. Login as User option will be used for user login page. Here user\_tab table is used to retrieve user ID, and password. There are restrictions as password incoorect, no user found (If the user ID is incorrect). Let’s see the code and query we used to make this page,

"select password from user\_tab where user\_id = %s"

If the query retrieves no user, we can be sure there is no user available with the entered user id. It prompts “No user Found” error.

Once the user id and password is retrieved, if the password mismatch with the entered password and retrieved password, then it prompts “Password incorrect” error.

If both user id and password is correct, verification will be successful and the page directs to “user\_home.py” module where user can access all his features.

3. Signup page redirects the user to add\_new\_user.py module where new user can be created by entering username, password, name, mobile number, and email address.

4. Exit option simply will close the application

**Sign up Page: (signup\_page.py)**

Signup page is where new users can create their own new profile to begin with the quiz application journey. Here user will be prompted to enter their new username, name, password, mobile number and the email address.

Here is the password mismatch with the password and re-type password, error will be prompted as “Password Mismatch”.

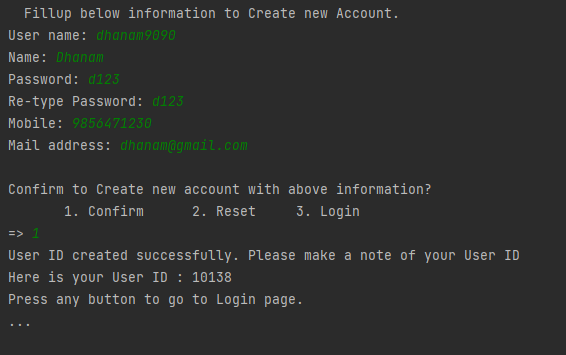
User\_id for the new user will be allocated automatically by getting the last user’s ID and incrementing it.

("select max(user\_id) from user\_tab")  
value = point.fetchall()  
for i in value:  
 max\_id = i[0]

User level and points will be allocated to the new user as user level = 1 and points = 0.

And finally new user will be created by the below query and the new user id will be shared with user,

"insert into user\_tab (`user\_id`,`user\_name`,`password`,`name`,`mobile`,`mail`,`user\_level`) Values(%s,%s,%s,%s,%s,%s,%s);"

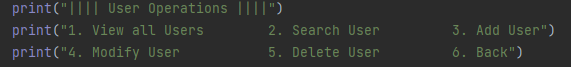


**Admin Homepage: (admin\_home.py)**

Here admin will have his/her features as User Operations and Quiz Operations.

**User Operations:**

User operations is where admin can view all users, search user, modify user details, delete user and create new user just like signup.



**View Users:**

Here admin can view all the users in quiz application using the below query,

query = "select \* from user\_tab;"

**Search User:**

This option will redirect admin to search\_user module where he will be presented with multiple search options.



Every option will use unique query to search user as below,

Search by ID:

"select \* from user\_tab where user\_id like '%s%'"

Search by username:

query = "select \* from user\_tab where user\_name like %s"

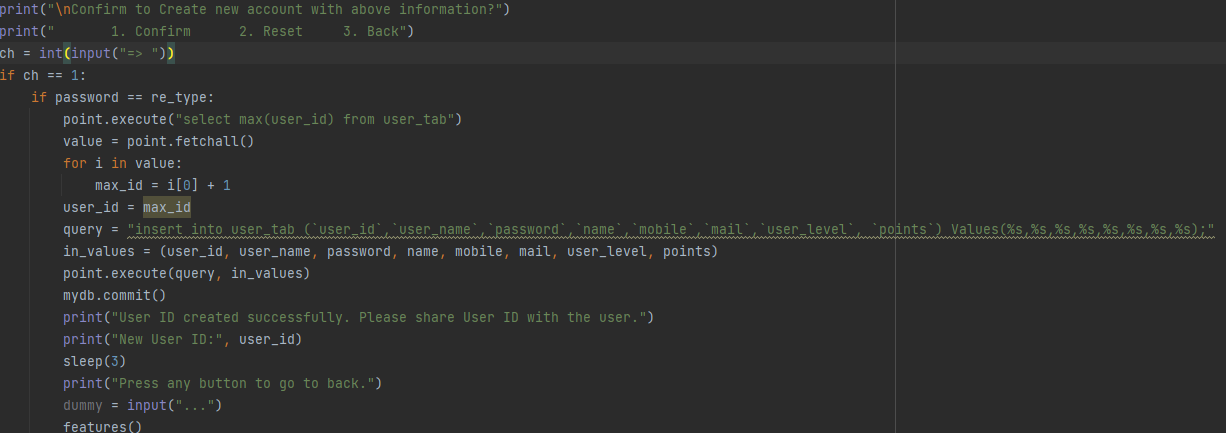
Search by Name:

query = "select \* from user\_tab where name like %s"

Search by Level:

query = "select \* from user\_tab where user\_level = %s"

**Add user:**

This page works just like Signup page. Once the admin entered all the information, and password is matched, he/she will be prompted as below, and continue the program as below,  


**Search User:**

This option will redirect admin to search\_user module where he will be presented with multiple search options.



Every option will use unique query to search user as below,

Search by ID:

"select \* from user\_tab where user\_id like '%s%'"

Search by username:

query = "select \* from user\_tab where user\_name like %s"

Search by Name:

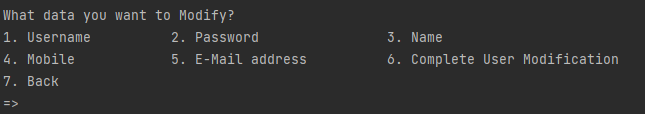
query = "select \* from user\_tab where name like %s"

Search by Level:

query = "select \* from user\_tab where user\_level = %s"

**Modify User:**

Here admin can modify any user’s name, password, mobile number email address or all the information. NOTE: Points and Level cannot be modified.



Modify Username: This will modify the username of a user by user\_id as per below query,

"update user\_tab set user\_name = %s where user\_id = '%s'"

Modify Password: This will modify the password of a user by user\_id as per below query,

"update user\_tab set password = %s where user\_id = '%s'"

Modify Name: This will modify the name of a user by user\_id as per below query,

"update user\_tab set name = %s where user\_id = '%s'"

Modify Mobile: This will modify the mobile number of a user by user\_id as per below query,

"update user\_tab set mobile = %s where user\_id = '%s'"

Modify Email address: This will modify the mobile number of a user by user\_id as per below query,

"update user\_tab set mail = %s where user\_id = '%s'"

Complete User Modification: This will modify all the information of user except level and points as per below query,

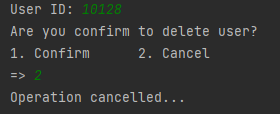
query = "update user\_tab set user\_name = %s, password = %s, name = %s, mobile = %s, mail = %s where user\_id = '%s'"

**Delete User:**

This option used to delete any user from the quiz application.

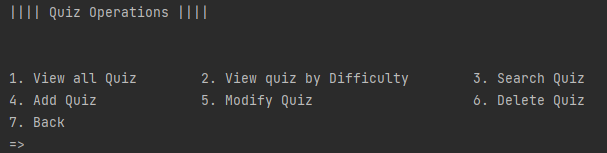
"delete from user\_tab where user\_id = %s"

To avoid deleting users by mistake, Admin will be prompted for a confirmation to delete the user as below,

****

In admin feature of user operations all the statements, codes, queries will be under try/except block to make sure there is no error, or crash in the application while operating with the application.

**Quiz Operations:**

****

Here admin will be presented with options as above to interact with Quizzes.

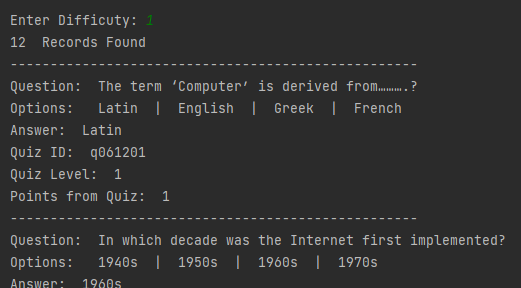
1. View All Quiz:

This option will be used to retrieve all the quizzes from quiz application database using the below query,

"select \* from quiz"

2. View Quiz by Difficulty:

Admin can retrieve quizzes by it’s difficulty just by entering the quiz level in this option.



Query:

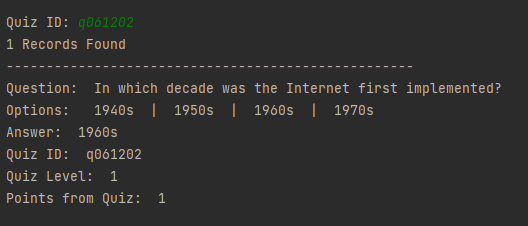
"select \* from quiz where q\_lvl = %s;"

3. Search Quiz:

Here admin can search quiz by it’s quiz id.

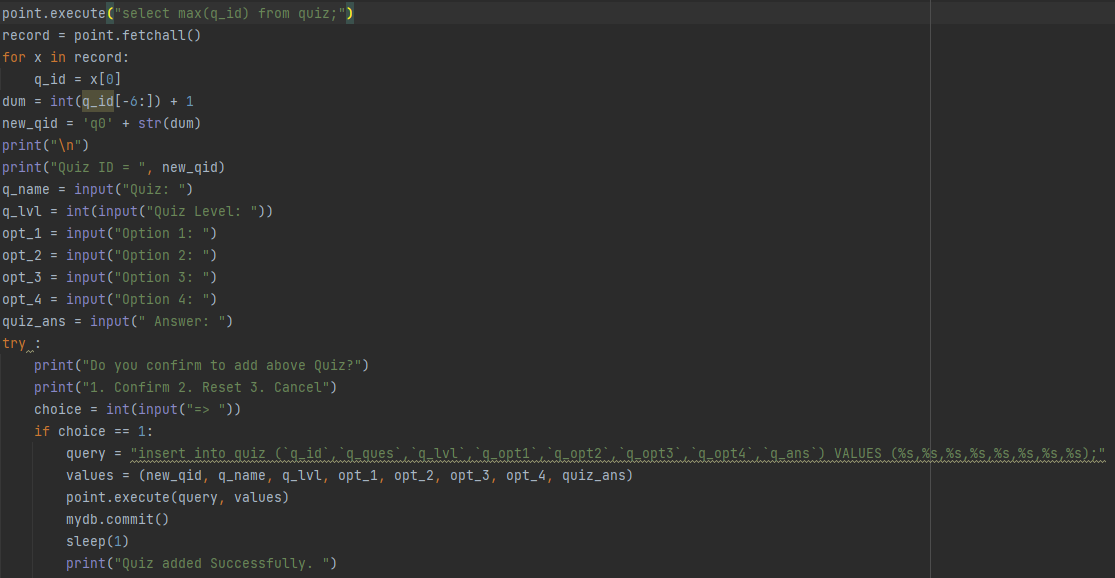
Query:

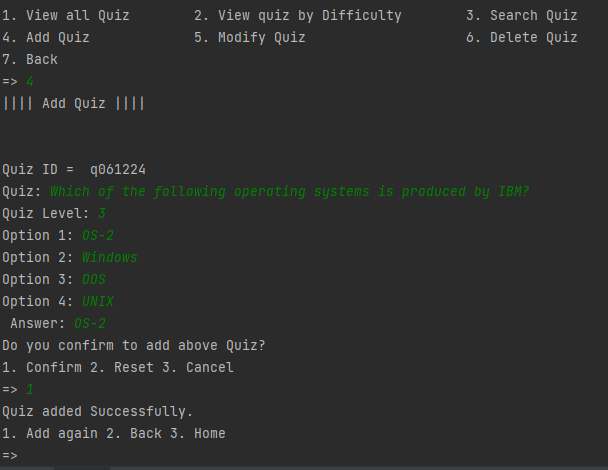
"select \* from quiz where q\_id = %s;"



4. Add Quiz:

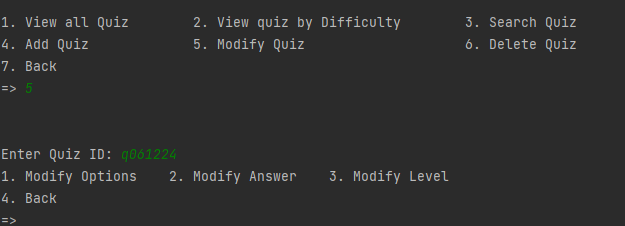
Here admin can add new quiz by entering question, difficulty level, options (1,2,3,4), and answer.

Quiz id will be allocated automatically by getting the last available quiz from database and incrementing it.  




5. Modify Quiz:

Here admin will be presented to modify quiz question options, answer and the difficulty level as below,



And the below queries used to modify quiz as per the selected option

1. Modify options:

"update quiz set q\_opt1 = %s, q\_opt2 = %s, q\_opt3 = %s, q\_opt4 = %s, q\_ans = %s where q\_id = %s"

2. Modify Answer:

"update quiz set q\_ans = %s where q\_id = %s"

3. Modify Difficulty Level:

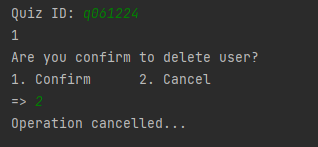
"update quiz set q\_lvl = %s where q\_id = %s"

**6. Delete User:**

This option used to delete any question from the quiz application.

"delete from quiz where q\_id = %s"

To avoid deleting users by mistake, Admin will be prompted for a confirmation to delete the user as below,

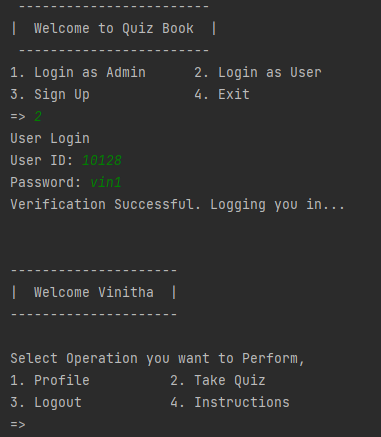


In feature of user operations all the statements, codes, queries will be under try/except block to make sure there is no error, or crash in the application while operating with the application.

**User Homepage: (user\_home.py)**

Once the user successfully verified their profile, and logged in, they will be redirected this user homepage to interact with the quiz application./

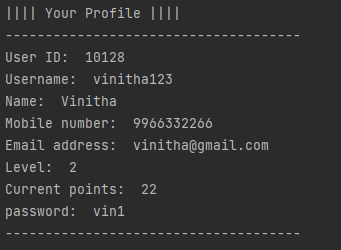
This page has features as Profile, Take quiz, logout and Instructions.



**1. Profile:**

In profile page, user can see their profile information such as username, user id, name, mobile number, email address and password.

For any changes in user information, user much reach administrator.



**2. Take Quiz:**

This option will redirect user to quiz home page where user can participate in quiz to test their knowledge and improve their skill.

There are two options to take quiz.

\* Start Quiz for your Level. (Recommended)

\* Start Quiz by Difficulty.

Start quiz for your level will retrieve the user’s level from database and match the level of question to make the quiz equal to their level.

For example, if the user is in level 1, and select this option, the below query used to retrieve questions from quiz table only those are in user’s level.

"select q\_id from quiz where q\_lvl = %s"

"select q\_ques from quiz where q\_id = %s"

Every quiz will get questions from table by random. So that user will not get question which was already answered.

Once the questions and options are retrieved from database, quiz will begin.

Once user answered all the questions, quiz will calculate the selected answers and match with the quiz answer column of quiz table to check the selected answers are correct or not using if statement.

"select q\_ans from quiz where q\_id = %s"

Each question will have 10 marks, and user must answer 7 or above questions to get pass marks as 70marks will be the qualification for passing the quiz.

If the user passed the quiz, they will get points as the quiz they took. For example if the user takes level 1 quiz, they will get 1 point for passing that quiz.

\* Level 1 Quiz -> 1 Point

\* Level 2 Quiz -> 2 Point

\* Level 3 Quiz -> 3 Point

\* Level 4 Quiz -> 4 Point

\* Level 5 Quiz -> 5 Point

And points will be updated to the points column of user table once they have passed the quiz.

"update user\_tab set points = %s where user\_id = %s "

There is another rule. User’s level will be improved when they cross 20 points.

\* Points 0-20 -> Level 1

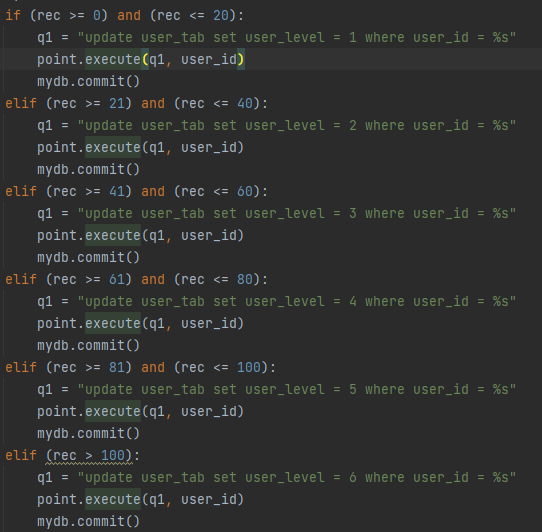
\* Points 21-40 -> Level 2

\* Points 41-60 -> Level 3

\* Points 61-80 -> Level 4

\* Points 81-100 -> Level 5

At the end of the quiz, if the user’s points crosses the level points, user’s level will be updated using as below,



Start Quiz by Difficulty will give 5 level of difficulties to start with the quiz.

\* Level 1 -> Beginner

\* Level 2 -> Amateur

\* Level 3 -> Comfortable

\* Level 4 -> Professional

\* Level 5 -> Master

Once the level is started, quiz will begin just as the previous start quiz feature and all the points validation and level validation will apply here as well.

**3. Logout:**

Logout option will close the application

**4. Instructions:**

Here user will have two options to pick,

\* User instruction

\* Quiz Instruction

**User Instruction:**

1. User can visit their Profile to check their points, level and personal information.

2. Any changes if required in personal information, user should reach Administrator.

3. To update password, user should reach Administrator.

4. User will get 0 points at the beginning of the journey.

5. User will be at Level 1 at the beginning of the journey

6. Points can be obtained as below,

\* Level 1 Quiz -> 1 Point

\* Level 2 Quiz -> 2 Point

\* Level 3 Quiz -> 3 Point

\* Level 4 Quiz -> 4 Point

\* Level 5 Quiz -> 5 Point

7. Level can be obtained as below,

\* Points 0-20 -> Level 1

\* Points 21-40 -> Level 2

\* Points 41-60 -> Level 3

\* Points 61-80 -> Level 4

\* Points 81-100 -> Level 5

\* Points 100 or above -> Mastered

**Quiz Instructions:**

1. There are five level of Quiz.

\* Level 1

\* Level 2

\* Level 3

\* Level 4

\* Level 5

2. Each level of Quiz will be harder as it increasing.

3. Quiz levels will be categorized as below in Quiz page.

\* Level 1 -> Beginner

\* Level 2 -> Amateur

\* Level 3 -> Comfortable

\* Level 4 -> Professional

\* Level 5 -> Master

4. Points can be obtained as below,

\* Level 1 Quiz -> 1 Point

\* Level 2 Quiz -> 2 Point

\* Level 3 Quiz -> 3 Point

\* Level 4 Quiz -> 4 Point

\* Level 5 Quiz -> 5 Point

5. Every Quiz will have 10 Questions.

6. Each Questions will take 10 marks, as the one total Quiz will have 100 marks.

7. Getting 70 or above mark is a Successful Quiz.

8. Successfully Quiz will give you points to your profile, based on the Quiz Level.

**Conclusion:**

Programming is the process of taking an algorithm and encoding it into a notation and implementing our plan to make it possible for a computer can understand and the user can have good experience and make use of it.   
This quiz book application created for education purpose where user can have simple quiz questions and answer them to improve their skill and knowledge. To add some fun for this process, we have added points, level and ways to improve them, so that user will get interested in achieving more level and earning more points which will naturally improve their skill.

We have seen what is Python, what is MySQP andWe have seen everything about this quiz application, and how it works.

As a creator of this application, I have experience so much joy while making this application. Thanks for taking time to check out my application and it’s about file.

Have a great coding!!!