INTRODUCTION

We are designing a quiz module with a proper interface. When the user starts attempting quiz, he will have choice about the questions category like which level of question he wants to attempt (Easy, Medium, Tough) . We will also keep record of user name, registration number and section. After completing all these required credential and level of question , he has to click on submit button to process further. After submission of required information, he has to attempt 10 questions either of easy, medium or tough, it depend what he selected on homepage.

The quiz is a application for to take test in an efficient manner and no time wasting for checking the paper. The main objective of quiz is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. For students they give papers according to their convenience and time and there is no need of using extra thing like paper, pen etc. This can be used in educational institutions as well as in

corporate world. Can be used anywhere any time as it is application(user location doesn’t matter). No restriction that examiner has to be present when the candidate takes the test.

According to us, this quiz is exactly the part of module we were looking for. But during some basic tests we found out, that it is not working very conveniently. However, the whole development of this quiz module with all its details is not feasible in the available time, even though 3 people are working on that project. Nevertheless the goal of the project is to write down the ideas and to create a concept of the quiz in a way that it could be understood easily. Therefore We came up with another idea, which can be realized by ourselves. in a feasible amount of time. So We decided to create a quiz module by ourselves. Net with the programming language python, because We are familier with this development environment and We are able to create this part within the available time.

Overview: This program lets you set up a series of questions. There is a flexible array of question types from which to choose. Questions can be given short titles that help you select which questions to retrieve when setting up a quiz. Test results can be viewed in various formats (either

with each student's paper listed separately, or in a table with answers in columns organized by question, and with aggregate results along the edges of each column and row). Blocks of data (coded or uncoded) can be extracted easily into spreadsheet programs for subsequent analysis.

Display Options:

- **Revisions:** The view screens let you review and revise questions, with no risk of losing existing response data.

- **Tabular Display:** You can decide how to organize the data: by student with a list of question titles and answers, or in a tabular form where each student's responses are in a row, and each answer constitutes a column.
- **Formatting:** A reversible truncation feature lets you tailor the data and improve the readability of a summary data table.

- **Omitting Responses:** The results pages let you hide and restore all answers from selected respondents, e.g. to omit data used in testing or data from unauthorized respondents.

- **Aggregate Results:** For questions with multiple options, the tabular display also reports aggregate numbers of responses for each answer option, which helps the instructor focus subsequent discussion on areas of common confusion.

- **Extracting Data:** The Create Data Table links generate unformatted blocks of data with column headings (separated by spaces), which can be copied and pasted into a spreadsheet program like Excel. The data blocks are generated two ways: using wording from selected answers and using the coded number of the response option, so coding is automatic.

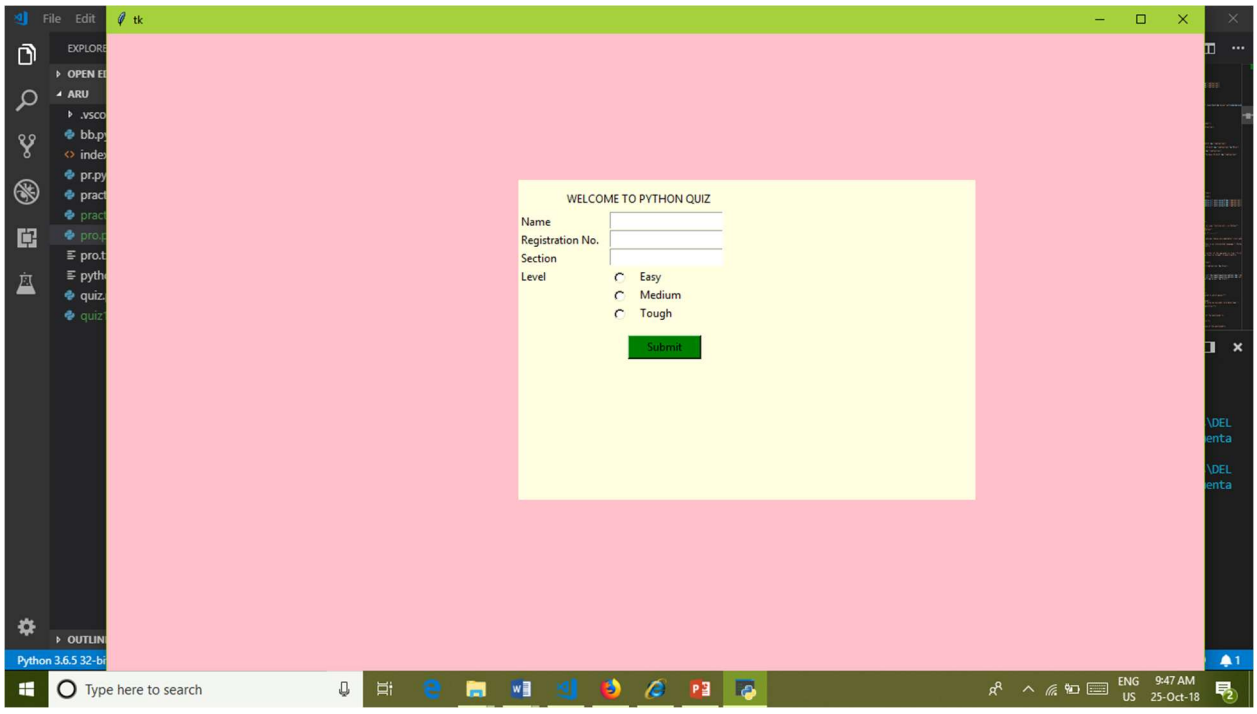
OBJECTIVES

- Aiding in language development with the questions based in the language alone
- Aiding in the team building process

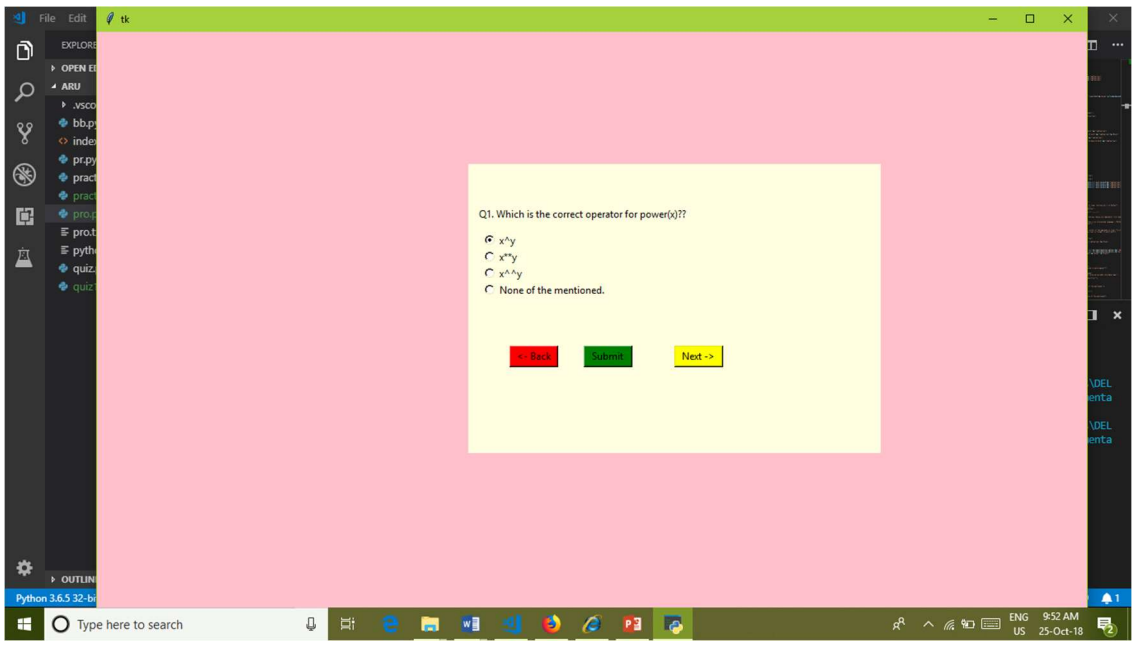
- Being fun to participate in but with a competitive element
- Provoking discussion and healthy debate amongst participating players / teams
- Aiding to relieve the tension of the daily work routine
- Promoting group harmony, whether in organisations, schools or just a group of friends
- Making new friends, from different cultures at times
- Likewise spelling bees improve the student's spellings and vocabulary
- Quiz competitions even help build the student's soft skills
- The questions in the quiz will be regarding python programming and same general questions related to computer science and technology with given four options.
- To increase the efficiency of and knowledge of the user related to computer science and programming.
- To provide a good and Knowledge full interface to the users to sharp their skills.

QUIZ MODULE DESCRIPTION

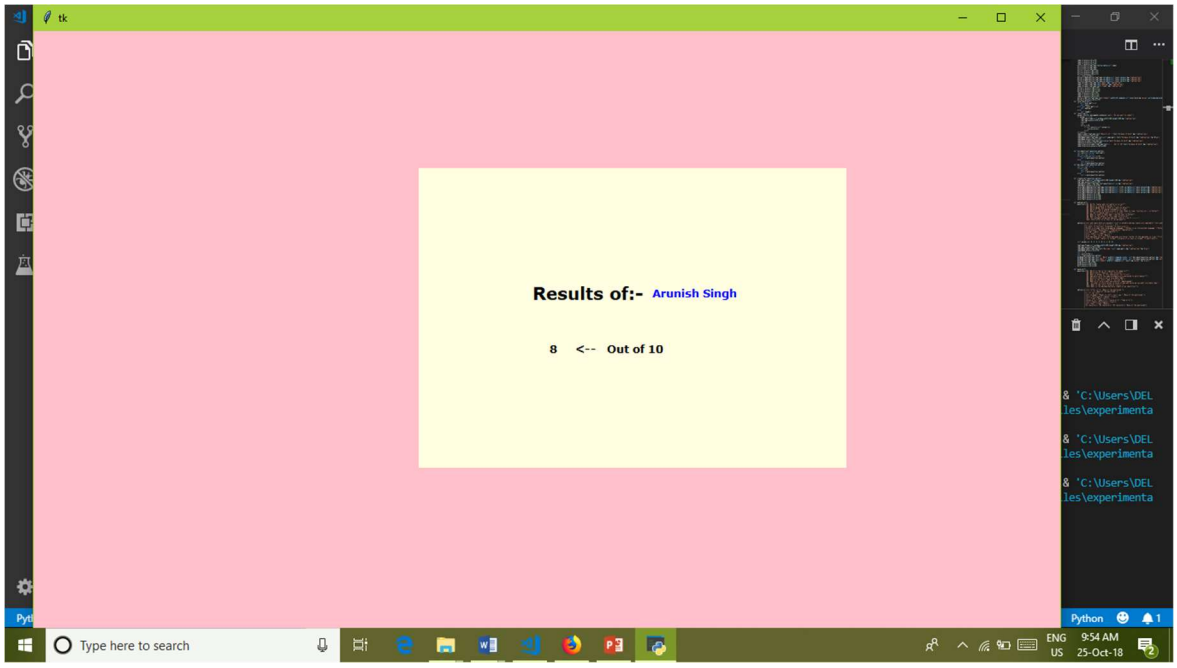
Home Page:- This page is the first page of our online quiz questions module. In this page the users will have to fill their name, registration number, section and difficulty level of question. This page has a submit button to move further for attempting questions.



Questions Page:- This page have the question according to the user's choice of difficulty level. The questions have four options to select a correct one. There are 20 question of each difficulty level (easy,medium and high). The page has a previous button to back to the previous question, a save button to save the question, a next button to go to next question and a End button to end the quizthat time instantly.



Result Page:- This page will have the result of the quiz attempted by the user with their name, registration number. This page also describes that how many questions you have attempted are correct and wrong. Overall it will show you about your performance in the quiz game.



CONCLUSION

- This quiz system module provides the service of attempting the quiz having different difficulty levels.
- The project is user friendly.It is very handy to use.
- The backend coding which is in python is easy to read and understandable.
- It displays the questions effectively and performs the function very effectively in the backend.

Conditions apply**