

WEB BASED QUIZ SYSTEM

RUBIX QUIZ

Nashtarin Nur
Student ID: 012153004

A Project in
The Department
Of
Computer Science and Engineering



Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

January, 2018

© Nashtarin Nur, 2018

Approval Certificate

This project titled “**Web Based Quiz System RUBIX QUIZ**” submitted by **Nashtarin Nur**, Student ID: 012153004, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on 13th January, 2018.

Board of Examiners

1.

Supervisor

Dr. Mohammad Nurul Huda

Professor and MSCSE Coordinator

Department of Computer Science and Engineering

United International University

2.

Examiner

Dr. Dewan Md. Farid

Associate Professor

Department of Computer Science and Engineering

United International University

Declaration

This is to certify that the work entitled “**Web Based Quiz System RUBIX QUIZ**” is the outcome of the research carried out by me under the supervision of **Dr. Mohammad Nurul Huda**, Professor and MSCSE Coordinator, Department of CSE, UIU.

Nashtarin Nur,
ID: 012153004,
MSCSE

In my capacity as supervisor of the candidate’s project, I certify that the above statements are true to the best of my knowledge.

Dr. Mohammad Nurul Huda
Professor and MSCSE Coordinator
Department of Computer Science and Engineering
United International University

ABSTRACT

The **RUBIX QUIZ** is a web based quiz system where students from class VI to X can attend the examination of multiple choice questions. In this platform they can also justify themselves after giving the examination as result will be come out after answering definite number of question for each set in each subject in each class. There is also option for admin who can edit questions and answers. Moreover, admin can add or remove questions also can remove a set. Here Django and Python are used at backend and HTML, CSS, JavaScript and Bootstrap at backend. Any user have to register himself or herself first to attend in the examination in RUBIX Quiz by giving a definite number of information like name, email address and password. When a new user is registered his or her information is saved in database. When he/she wants to login in the site they have to put the registered email address and password and it is being authenticated and after that logged on successfully.

Acknowledgement

Praise is for Almighty Who showered His kind blessings in during this Project.

It has been great pleasure for me to develop the “**Web Based Quiz System RUBIX Quiz**”. I have gathered sufficient knowledge and experience during this project.

I would like to thank my honorable teachers and supervisor of this project **Dr. Mohammad Nurul Huda**, Professor, Department of Computer Science and Engineering, United International University, who guided me to proper analysis of the system and helped to develop an elegant and efficient system. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project. I must also convey my special thank and gratitude to all of respected teachers of my department. I would like to thank to all of my friends for their valuable suggestions and assistance for completing.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

TABLE OF CONTENTS

CONTENTS	PAGE
CHAPTER 1: INTRODUCTION	9-11
1.1 : Introduction	9
1.2 : Background	9
1.3 : Web/Internet	9
1.4 : Quiz System	10
1.5 : Web Based Quiz System	11
CHAPTER 2: BACKEND AND FRONTEND	12-13
2.1: Introduction	12
2.2: Python	12
2.3: HTML	12
2.4: CSS	13
2.5: Bootstrap	13
2.6: Django	13
CHAPTER 3: SYSTEM ANALYSIS	14-18
3.1: Introduction	14
3.2: Basic Block Diagram	14
3.3: Database	15
3.4: Users	16
3.5: Functions	16
3.6: Technologies Used	16
3.7: Dataflow Diagram	17
3.8: Advantages	18
3.9: Conclusion	18
CHAPTER 4: SYSTEM DESIGN	19-25
4.1: Introduction	19
4.2: MVC of RUBIX Quiz	19
4.3: System Design	20
4.4: Working Principle	21
4.5: ER Diagram	24

4.6: Conclusion	25
CHAPTER 5: EMBODIMENT AND EXPLORATION	26-36
5.1: Introduction	26
5.2: Commencement	26
5.3: Implementation	27
5.4: Conclusion	36
CHAPTER 6: CONCLUSION AND FUTURE WORK	37-38
6.1 Introduction	37
6.2 Selection of Wrong and Correct Answers	37
6.3 Explanation of Wrong Answers	37
6.4 Timestamps	37
6.5 Secure Admin	38
6.6 Conclusion	38
REFERENCE	40
APPENDIX	41-43
Some Sample Source Code of the Project	41

LIST OF FIGURES

FIGURES	PAGE NO
Fig 3.1: Basic Block Diagram	14
Fig 3.2: Tabular Form of Personal Information	15
Fig 3.3: Tabular Form of Question Bank Database	15
Fig 3.4: Dataflow Diagram	17
Fig 4.1: MVC Diagram	19
Fig 4.2: Workflow Diagram of Admin	20
Fig 4.3: Workflow Diagram of Student	20
Fig 4.4: Signing in from User's View	21
Fig 4.5: Registration	22
Fig 4.6: Online Test	23
Fig 4.7: ER Diagram	24
Fig 5.1: Staring the site in PC	26
Fig 5.2: First Appearance of RUBIX Quiz	27
Fig 5.3: Admin Login	28
Fig 5.4: When the Admin Logged On	29
Fig 5.5: Admin Works	30
Fig 5.6: Editing Questions and Sets	30
Fig 5.7: New User's Form	32
Fig 5.8: Login as an User	33
Fig 5.9: Notification of Successful Login	34
Fig 5.10: Choice of Set	34
Fig 5.11: Appearance of Questions	35
Fig 5.12: Checking Answers	35
Fig 5.13: Showing Result	36

Chapter 1

Introduction

1.1 Introduction

The education system are getting digitalized day by day in Bangladesh. In nearly 15 years ago the HSC and SSC were published in newspaper. But now days these public exams results are publishing on internet. So, for pacing up with the world and also requirements of students the education system are getting more reliable on internet. The **RUBIX QUIZ** is a such platform for students to make their life more easy acquiring academic education. This web based quiz system make them justify themselves for their final examinations. Moreover, this can make a new era in education system of Bangladesh.

1.2 Background:

Previously the education system of Bangladesh depends on elaborate answers of questions rather than one line answers .But now time has been changed. In HSC, SSC,JSC examinations of Bangladesh multiple choice questions are mandatory. Nearly half of the exam marks are determined for these multiple choice questions. So these type of questions carry a major marks to obtain a good result. It is very much need for the students to have some software or website related to these questions where they can practice properly. RUBIX Quiz is such a platform where students can judge themselves to a new level.

1.3 Web/Internet:

The **Internet** is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists

of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing. As the Internet is a heterogeneous network, the physical characteristics, including for example the data transfer rates of connections, vary widely. It exhibits emergent phenomena that depend on its large-scale organization.

1.4 Quiz System:

Now days Bangladesh education system relies on quiz system. Quiz system means the examination on multiple choice questions. In this process student has four choices for each question. Among these one question is correct other four are wrong. Student have to select the right answer to get the marks. If the student mark the any of wrong answer the marks will be zero for this questions. Mainly examiner tries to give the choices of similar pattern to the correct answer so that the student get confused. But who studies hard moreover heart and soul only he/she can answer each question correctly. So this type of examination system imposes students to study thoroughly. Moreover, in this system the student have to memorize less but have to study attentively. This type system lessens the pressure of brain for the students. Previously this type of system greatly relied on humans rather than computers. The teachers made the question papers with four answer choices. Students had to sit for the examination within definite time. In this certain period they have to answer a significant number of questions according to the academic curriculum. They had to fill up OMR sheet to answer the questions. For each question answer should be one. One round shaped box must be filled up with ball pen. If someone fill up round box more than one the question answer will be treated as a wrong answer.

Now days quiz system are going to be handle more on computer rather than manually. Students have to give exams on the internet or software related to quiz system. This type of sites or software have definite number of questions. After choosing the correct answers they have to submit the answers and the result and marks will come out. This type of system is best for making best use of time. Student don't need to fill up the OMR sheet. They just have to click on the correct answers. This saves a lot of time for the students.

1.5 Web Based Quiz System

Web based quiz system is a system where students can give examination of multiple choice question on internet or web. They don't have to move to other places to check out their preparation for final exams. They just need an internet connection and such website where he/she can easily justify their preparation for examinations. RUBIX Quiz is a such web based quiz system where student from classes VI to IX can take examinations of multiple choice questions. Admin can add, delete or edit questions and sets for each subject within each class. In such system a student has to register first to the website. If they are registered successfully they can easily prepare themselves for the final examinations for each subjects. The main advantage of this system is that student don't have to move to the coaching centers to prepare themselves perfectly. Moreover, if they can contact with the admin of the website admin can provide them with more questions to solve and also more set to practice. As a result this system can contribute a lot for the preparation for the students of upcoming examinations.

Chapter 2

Backend and Frontend

2.1 Introduction:

RUBIX Quiz web based quiz system mainly used Python as a programming language. It also used HTML, CSS, BOOTSTRAP and JavaScript. Moreover Django is used as database. It means Django works at backend and HTML, CSS, BOOTSTRAP and JavaScript works as frontend.

2.2 PYTHON:

Python now-a-days has become very much popular among the programmers. It's a high-level programming language. This language is created by Guido van Rossum. The first releasing time of python is in 1991. It let them the coders experience a new era of programming that coders or programmers can code programs only with few lines rather than C, C++ or Java.

Python has automatic memory management. Moreover, it has object oriented feature, functional features too. These are things that made python to a signified level. Python is available for almost all operating system. It is a open source software which also has community based development model. Python software Foundation is a non-profit foundation who manages the Python.

2.3 HTML:

HTML is the abbreviation of **Hypertext Markup Language** which is mainly used for making webpages and web application. With the collaboration of Cascading Style Sheets (CSS) and JavaScript it forms a platform for World Wide Web. HTML is received by web browsers. HTML mainly depicts the structure of web browser and also the appearance of the webpage. HTML have the capability to embed programs written in a scripting language like JavaScript which can control the contents of the web pages. Moreover CSS is the inclusion which depicts the looks and layout of the page.

2.4 CSS

The elaboration of CSS is **Cascading Style Sheets**. It describes the presentation of document written in HTML. Combined with JavaScript and HTML, CSS represents the webpages properly. The main theme of CSS is to decorate the webpage. CSS works to make layout, give colors and fonts. Also this can be controlled with HTML by giving extension .css file and it also reduces and repetition in the proper structural content.

Only changing the CSS the outlook can be changed overall webpage without interrupting markup language. There are various sources for the CSS information such as web browser, the user and the author. The information of CSS can be separated as document or embedded into an HTML. Moreover, different styles depends on which output will be used. The process cascading is used to select display with CSS with higher to lower priority.

2.5 BOOTSTRAP

Bootstrap is mainly concerned with front end development. It is an open-source front end web framework. It consists of HTML and CSS-based design templates for forms, buttons, navigation and interface components.

Bootstrap was developed as a framework to maintain consistency across internal tools. Before Bootstrap inconsistencies are major burden for developers. HTML structure, CSS declaration and some cases also mixed with JavaScript code. HTML components are provided with a set of style sheets which depicts the basic style definitions.

2.6 Django

Django is known as an open-source free web framework that is written in Python. Django always follows MVT means Model, View and Templates pattern. DSF means Django Software foundation maintains it which is a non-profit platform. The main target of Django is to ease the complex database driven websites. It imposes on “plug ability” of components and the principle which actually has negativity with repetition. Here Python is used models of data and also for the setting of files. Create, read, update and delete interfaces are provided by Django.

However, the core Django can be seen in MVC means Models, Views and Controller architecture. There are several core frameworks such as framework for developing GIS applications, creation Google site map tools. Moreover it helps to configure third party code to be plugged into a regular project

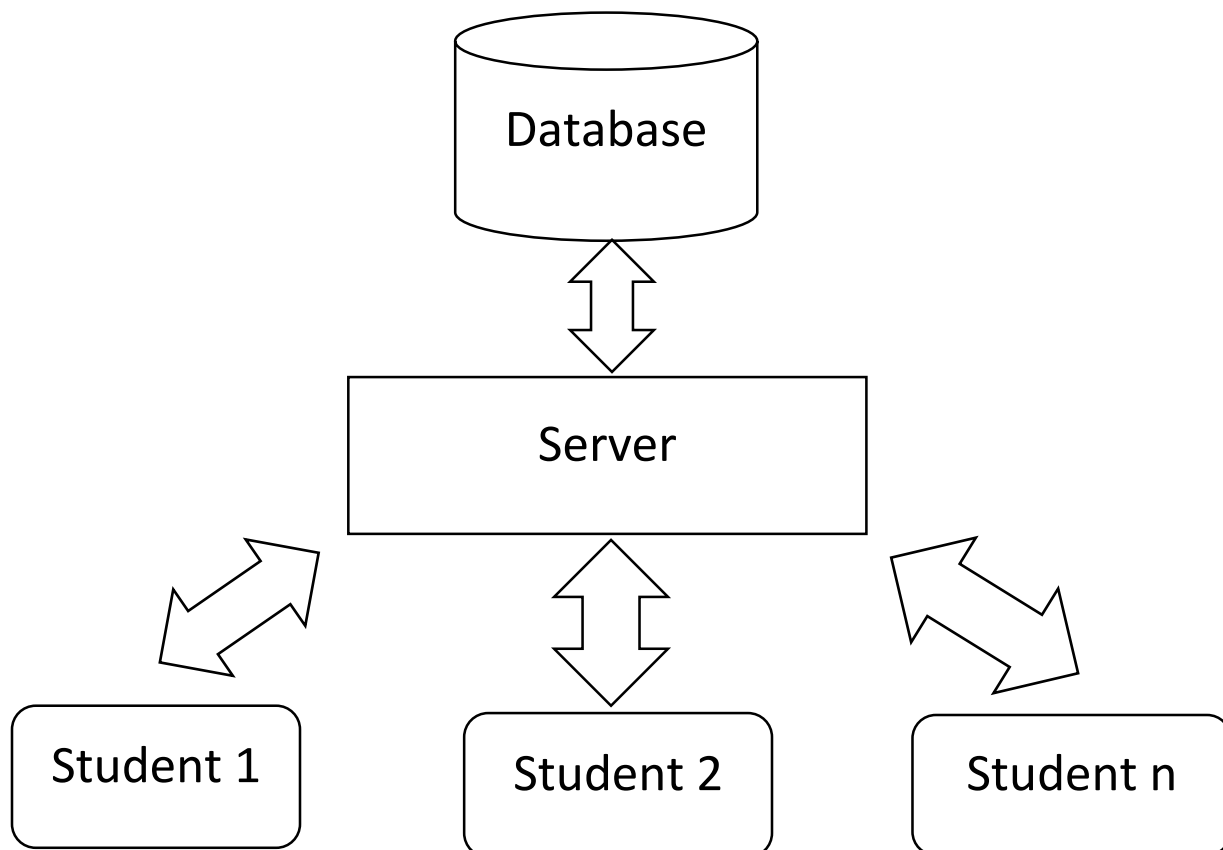
Chapter 3

System Analysis

3.1 Introduction:

RUBIX Quiz is mainly developed by the python language. Python language is much more flexible and easy and smart language than other like PHP or C++.It can easily adapted the what a programmer wants. The codes of Python are less than others as it is smart build in language. Moreover HTML is used to code the webpage. This is the most popular language for web page development. Moreover CSS is used to design and enhance the beauty and appearance of the page.

3.2 Basic Block Diagram:

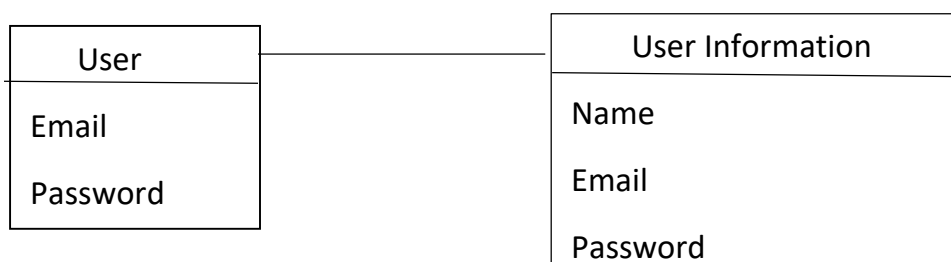


3.1 Basic Block Diagram

In database all questions and related answers are saved. Students have to register first to this server. The number of student can join in this site are not limited. Any student can register here but the students from class six to ten will be most benefited from this site.

3.3 Database:

In RUBIX Quiz Django Database is used. This database is working on computer hard disk for now rather than cloud. The questions and choices are saved in the Django database. The tabular form of database is depicted below:



3.2 Tabular Form of Personal Information Database

Question_Bank
Question
Option1
Option2
Option3
Option4
Correct Answer

3.3 Tabular Form of Question Bank Database

Here in RUBIX QUIZ Django Database is used. Mainly as it is not launched to any server yet the database is being filled in hard disk of this pc. Until the Database are filled admin can add as much as questions he/she wants. The final approach should be like that the RUBIX quiz will be connected to server mainly it will then use the cloud storage. Moreover the user information for each user will be saved in this server.

3.4 Users:

In this RUBIX Quiz the users are mainly the students. The students from classes six to nine. The students have to first registered to the website. Then they can go to the questions to answers. After they answered all the questions they can check their result too.

3.5 Functions:

3.5.1 Server End:

The functions of the server end are such that, Firstly have to register to the RUBIX Quiz if the user appear here for the first time. Then the another task of server end to authenticate the existing users. At last the most important task is to route the information from the source client to appropriate destination client.

3.5.2 Client End:

The functions of the client end are such that its work is to instruct the users who want to register. Moreover, it has to provide proper interface to sign-in, login and test.

3.6 Technologies Used:

Here the technologies are used depicted below

Backend:

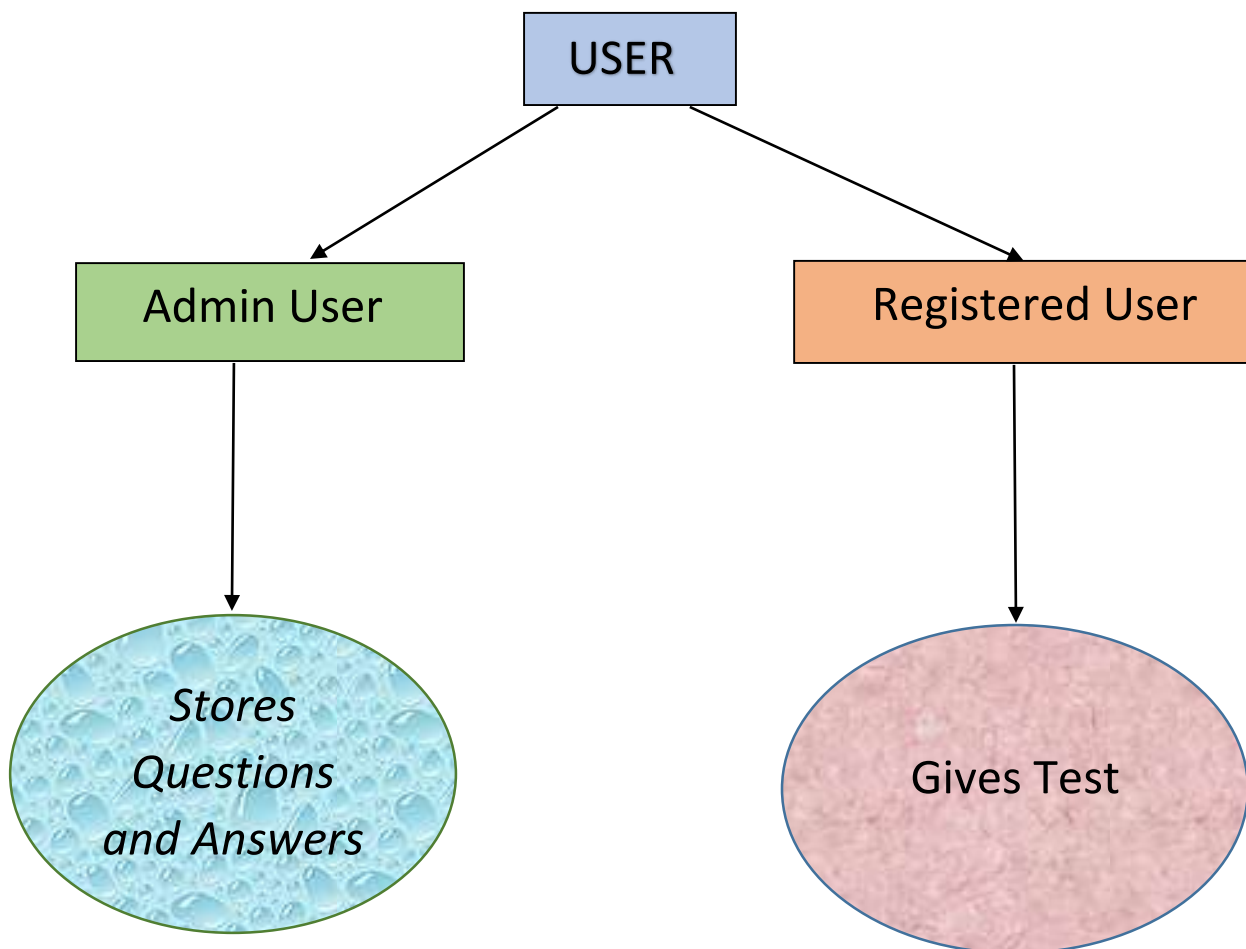
Django

Frontend:

i) Bootstrap

- ii) HTML
- iii) CSS
- iv) JavaScript
- v) Python

3.7 Dataflow Diagrams:



3.4 Data Flow Diagram

3.8 Advantages:

- RUBIX Quiz allows registers users to take test and view their result online.
- Here users can take each test several time as they want
- This provide a platform where user can choose from variety of aptitude test

3.9 Conclusion:

This RUBIX Quiz is very beneficial for students who want to increase the accuracy of their Preparation. They can endure knowledge and their intellectual ability to know their preparation well increases in a rapid race using this application. Moreover this project gives me a clearer understanding of real world application development.

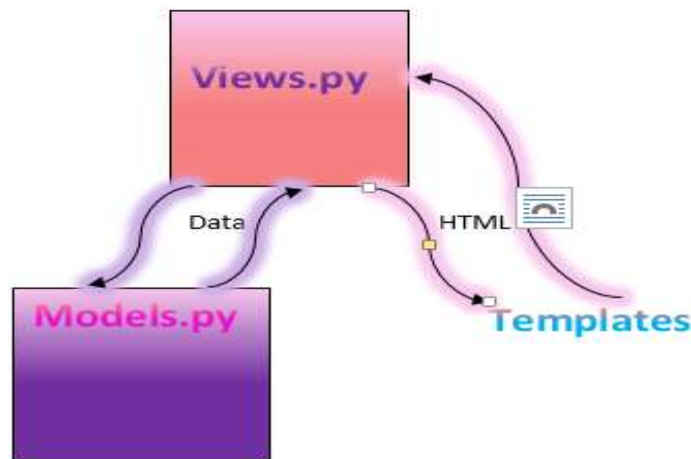
Chapter 4

System Design

4.1 Introduction:

RUBIX Quiz is an intelligent website to justify the student preparation. In this chapter we will discuss about the workflow of this project. Here, the admin can add or delete or edit questions. Admin can create several sets for each subjects. Each subjects has several sets. In each set student can attend a number of questions to answers. Admin has to login of the website with definite password.

4.2 MVC of RUBIX Quiz:



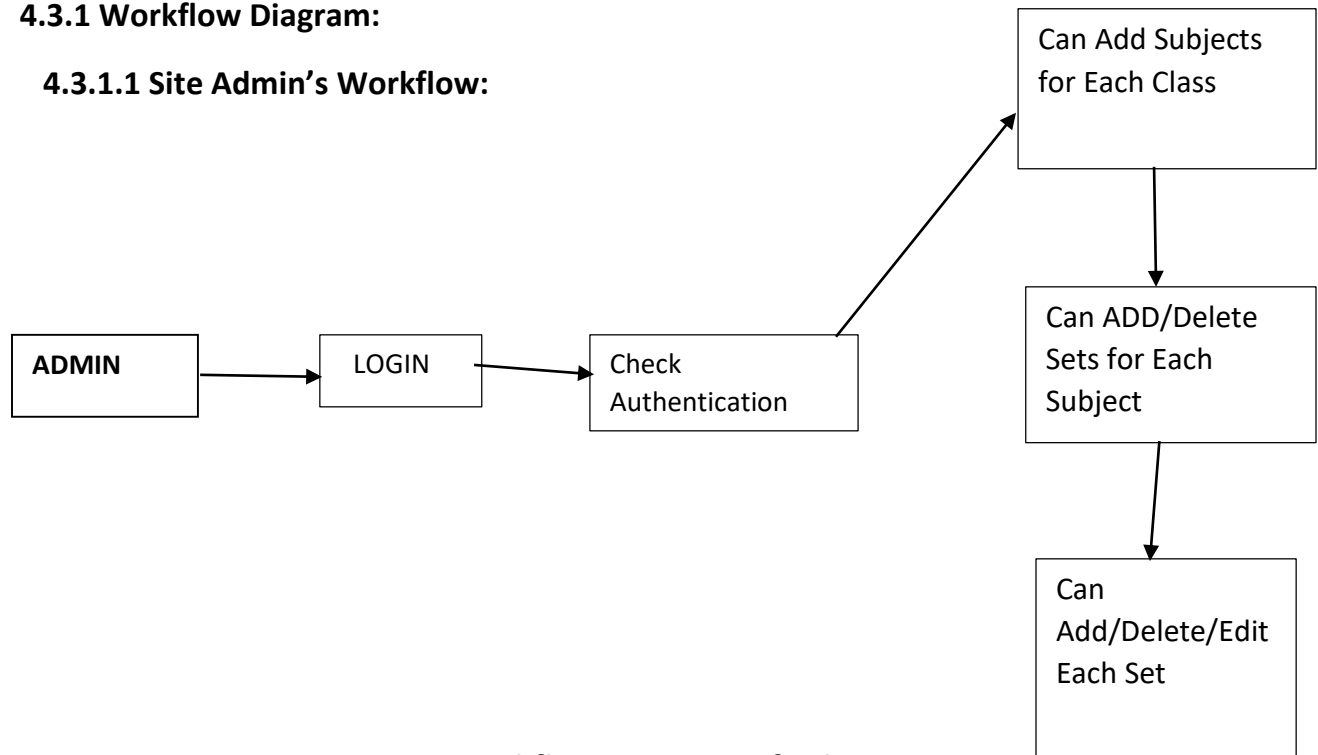
4.1 MVC Diagram

MVC stands for models, view and control. Models and views share data which are controlled by controller. Views share templates to show the website page through HTML. This is how the main program works in the RUBIX quiz

4.3 System Design:

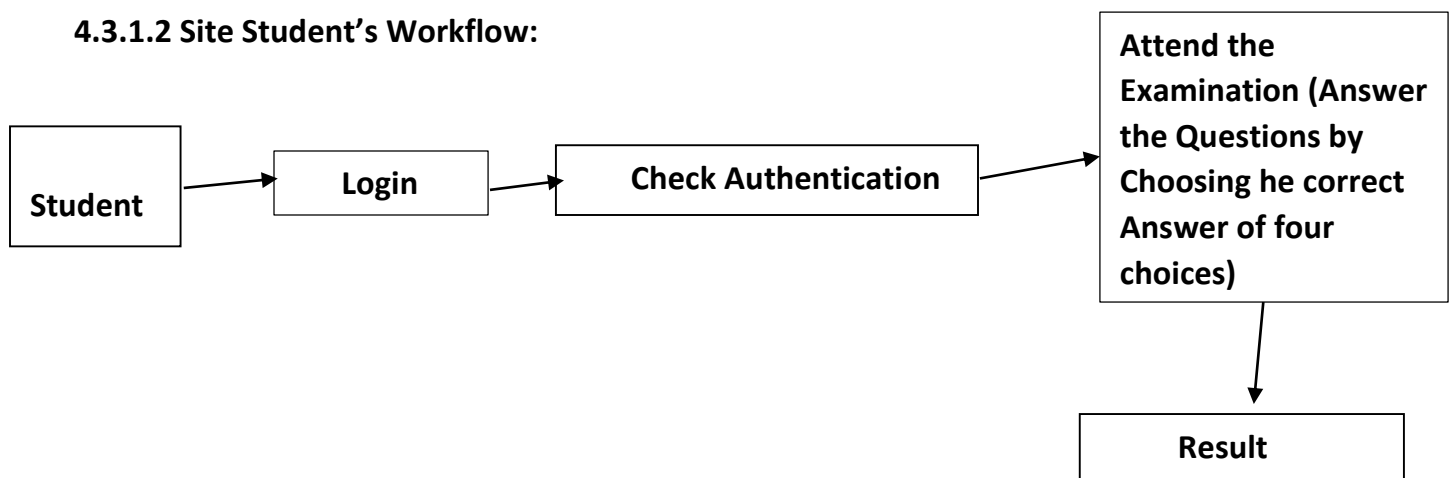
4.3.1 Workflow Diagram:

4.3.1.1 Site Admin's Workflow:



4.2 Workflow Diagram of Admin

4.3.1.2 Site Student's Workflow:



4.3 Workflow Diagram of Student

4.4 Working Principle:

4.4.1 Developer's View:

4.3.1.1 Database Creation:

There are two table and these will be

- Question_Bank Table
- Personal_information Table
- Storing questions and answers in the database within the Question_Bank table
- When the user will register here for attending the examination the details of the user will be stored in the personal_information table
- The new student have to be registered here to give the tests.

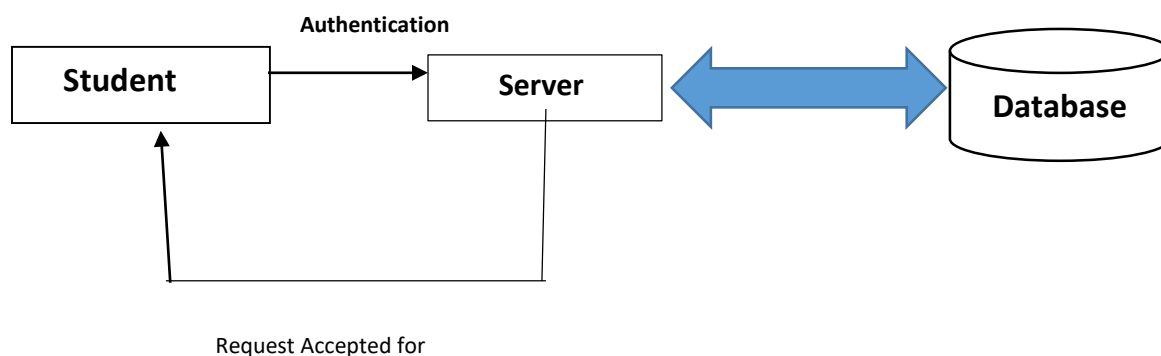
4.4.1.2 Logic:

At the time of registering the site the details of the user will be validated

- The given answers will be tested from the database and the result will be calculated according to the right answers. At last the result will be published on the screen.

4.4.2 User's View:

4.4.2.1 Sign In

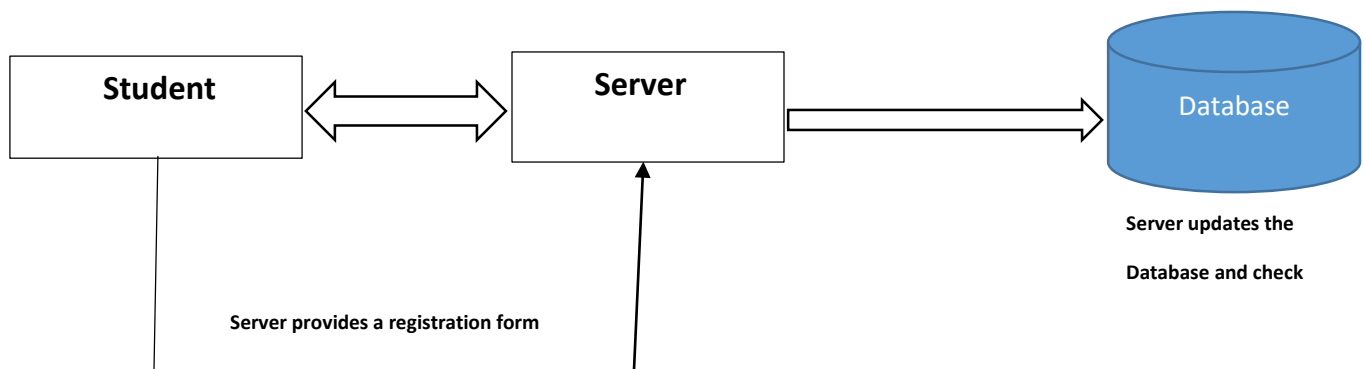


4.4 Signing in from User's View

- User request for authentication for server
- The server starts to check the database

- After that server star to modify the user
- After a successful authentication, the server initialize the student.

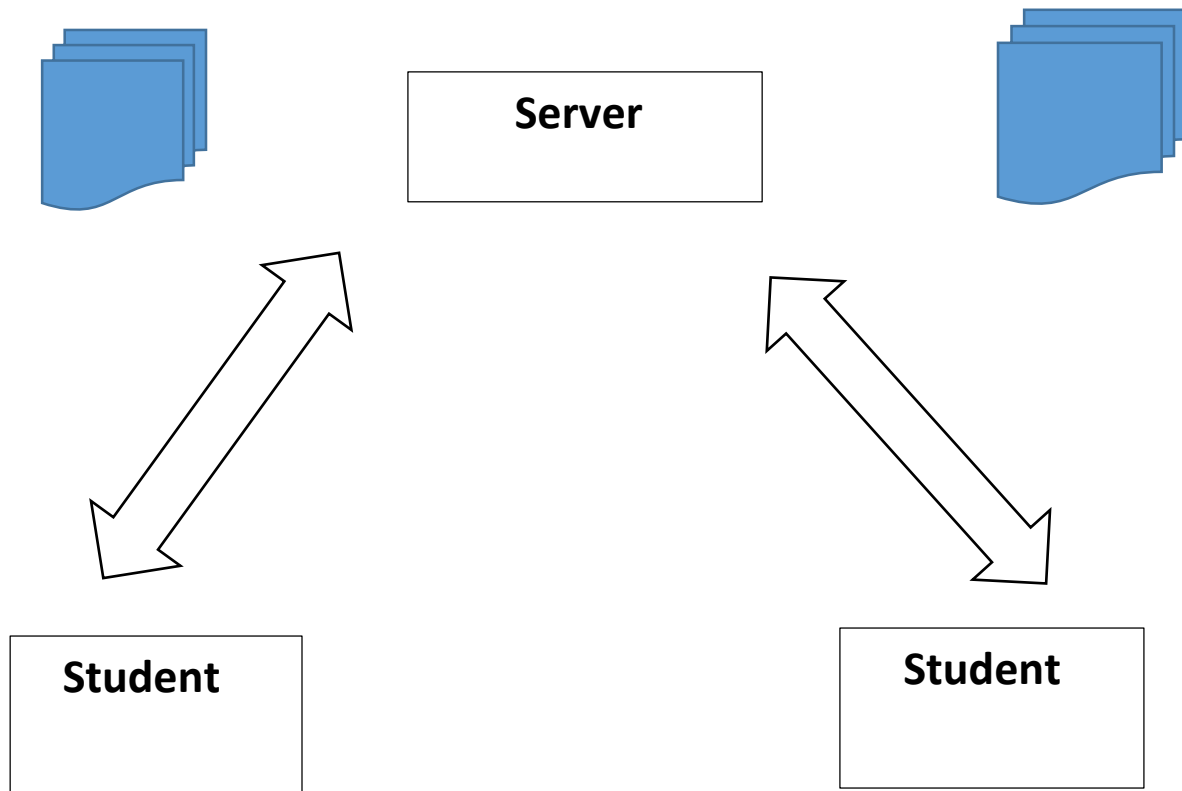
4.4.2.2 Registration:



4.5 Registration

- Users can apply for a registration form from the server to register.
- Here they have to fill up the student's information.
- They have to fill up the name, email and to give a password and confirm it
- Then the server update the database.
- After updating the database the server check the information with database
- Completing the checking of database the server authenticate the student or user with the definite registration information
- Admin also have to login to be authenticated
- After successful login of the admin, he/she can add ,remove or delete
- Again, a successful authentication of the student with the server after registration student can appear for the exam

4.4.2.3 Online Test:



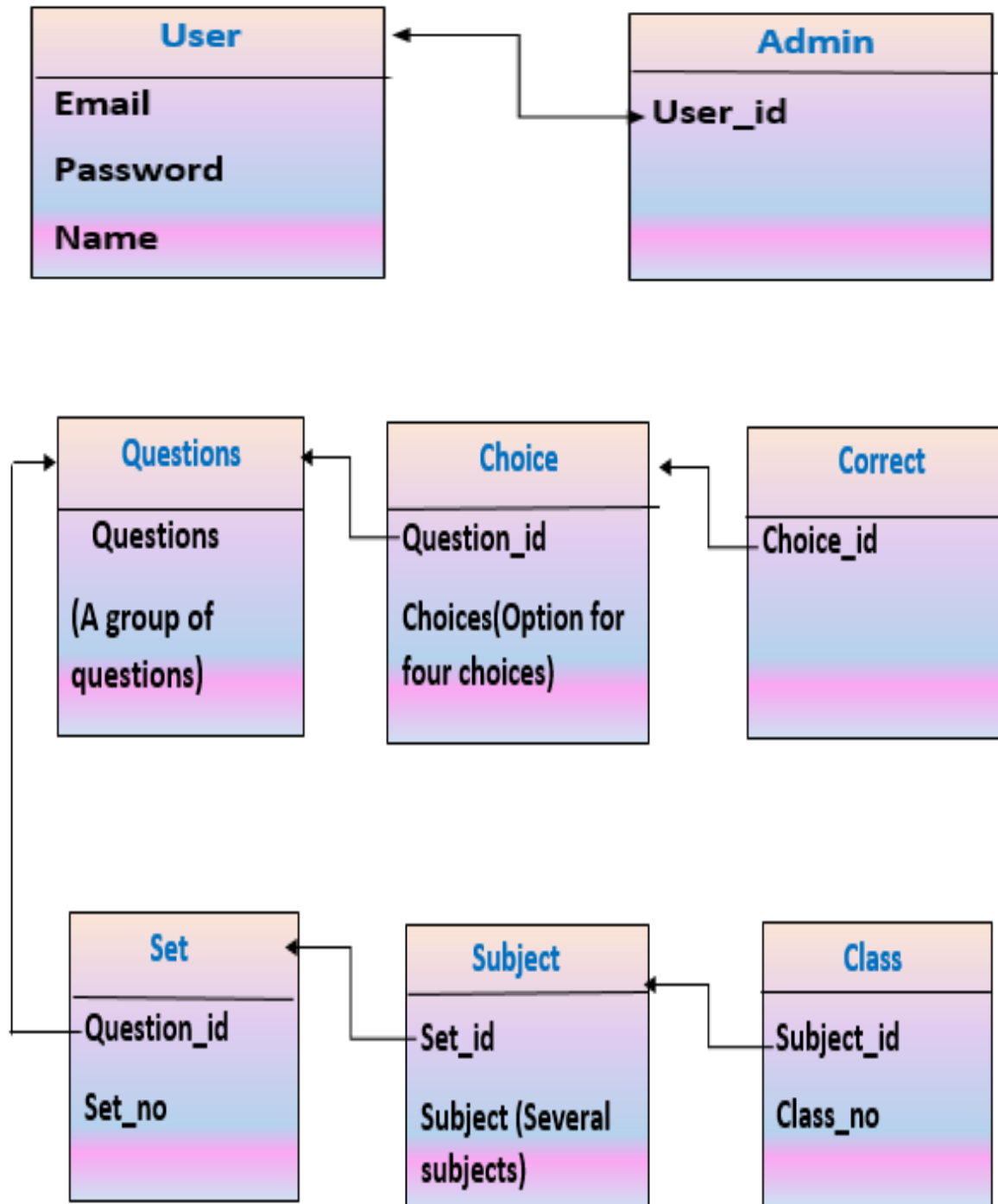
4.6 Online Test

- The person can access the details whoa is authorized
- The student can login by giving email id and password
- Admin can add/delete or edit question and answers
- Admin is responsible for updating the site

4.4.2.4 Test Analysis:

The student gets the result of the test. He/She can see the marks of him/her out of total for a definite set of each subject.

4.5 ER Diagram:



4.7 Entity Relationship Diagram

4.6 Conclusion:

In this chapter we elaborately describe about the working principle of RUBIX Quiz. Here, we discuss about the registration, authentication, online test and test result. Moreover, this chapter depicted the ER diagram and MVC (Model, View and Controller) of RUBIX Quiz. Eventually we have come to know that database is updated with every registration of every student and this database authenticate with student when a student wants to appear the online test in the RUBIX Quiz.

Chapter 5

Embodiment and Exploration

5.1 Introduction:

In this chapter we will demonstrate how the project is implemented and worked. At first we have to install Django to work as a database in this pc. Later on we will promote it to the server. Moreover, it's depicted in this chapter how RUBIX Quiz works and how can someone operate it. Users will find it more ambient, easy handling and more comfortable to use or to appear exams and see the results.

5.2 Commencement:

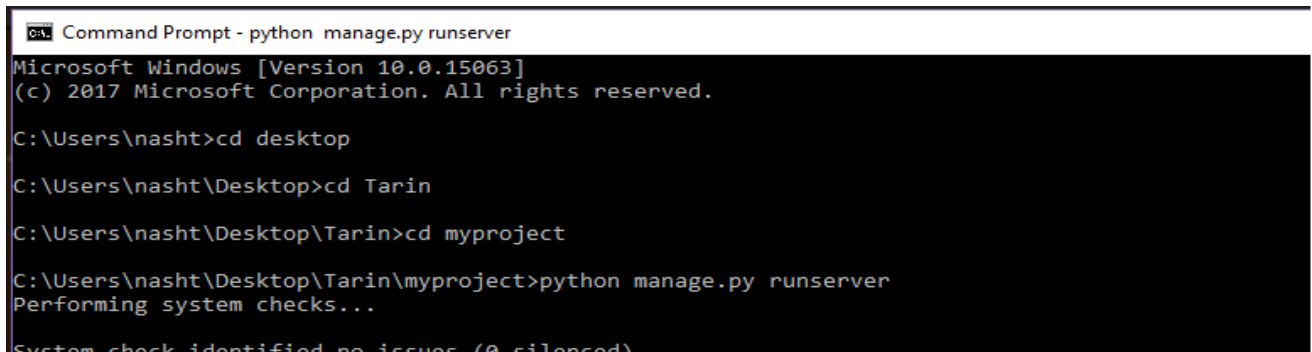
At first the project must be launched on the cmd. The commands of the cmd are

cd file location

Then have to write that

Python manage.py runserver

After that the program will start to work on the computer and the site will be displayed on the browser. The site in the browser will be localhost as the project has not been promoted to the server yet. The commencement of the project is shown below:



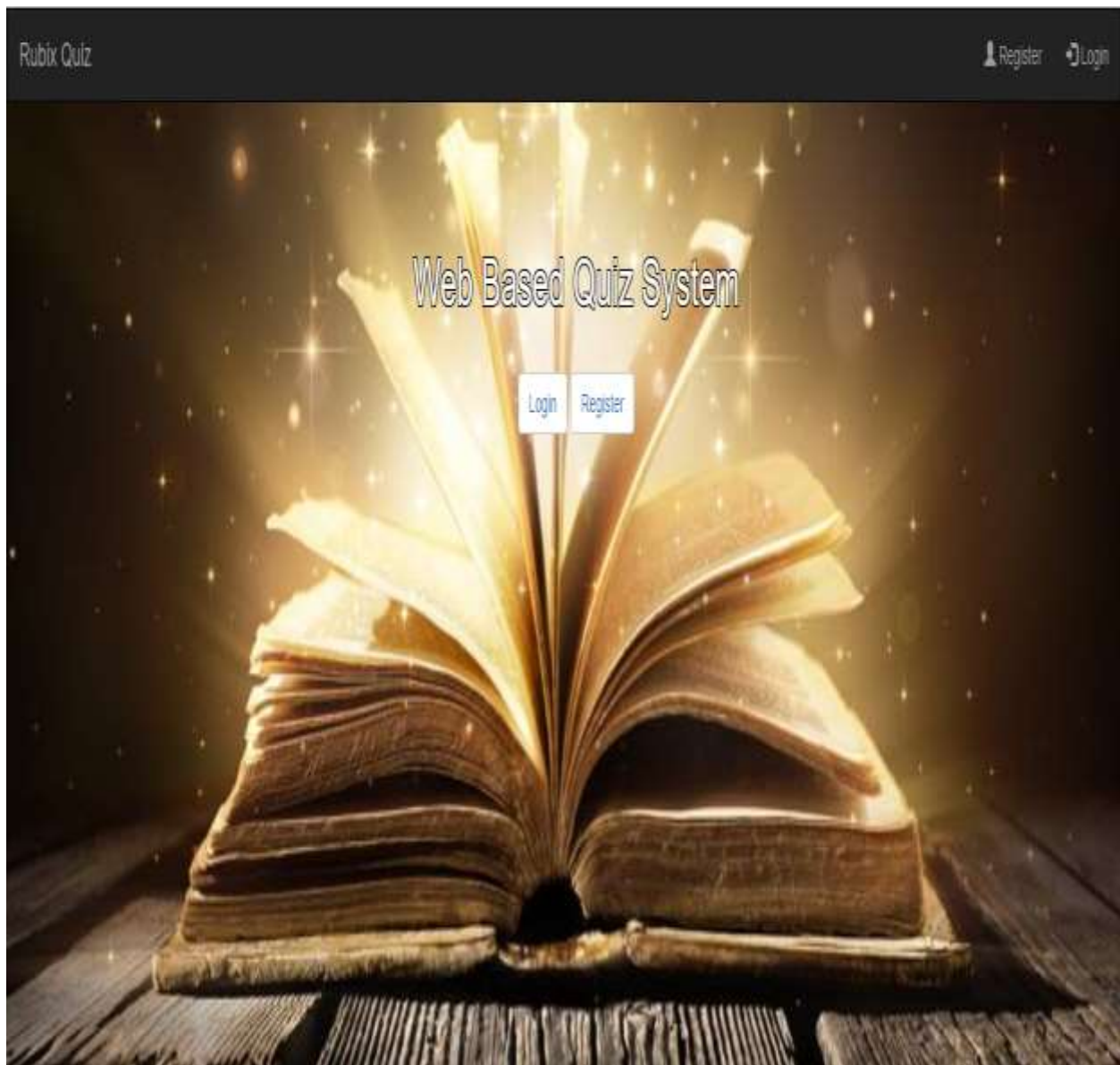
```
Command Prompt - python manage.py runserver
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\nasht>cd desktop
C:\Users\nasht\Desktop>cd Tarin
C:\Users\nasht\Desktop\Tarin>cd myproject
C:\Users\nasht\Desktop\Tarin\myproject>python manage.py runserver
Performing system checks...
System check identified no issues (0 silenced)
```

5.1 Starting the site in PC

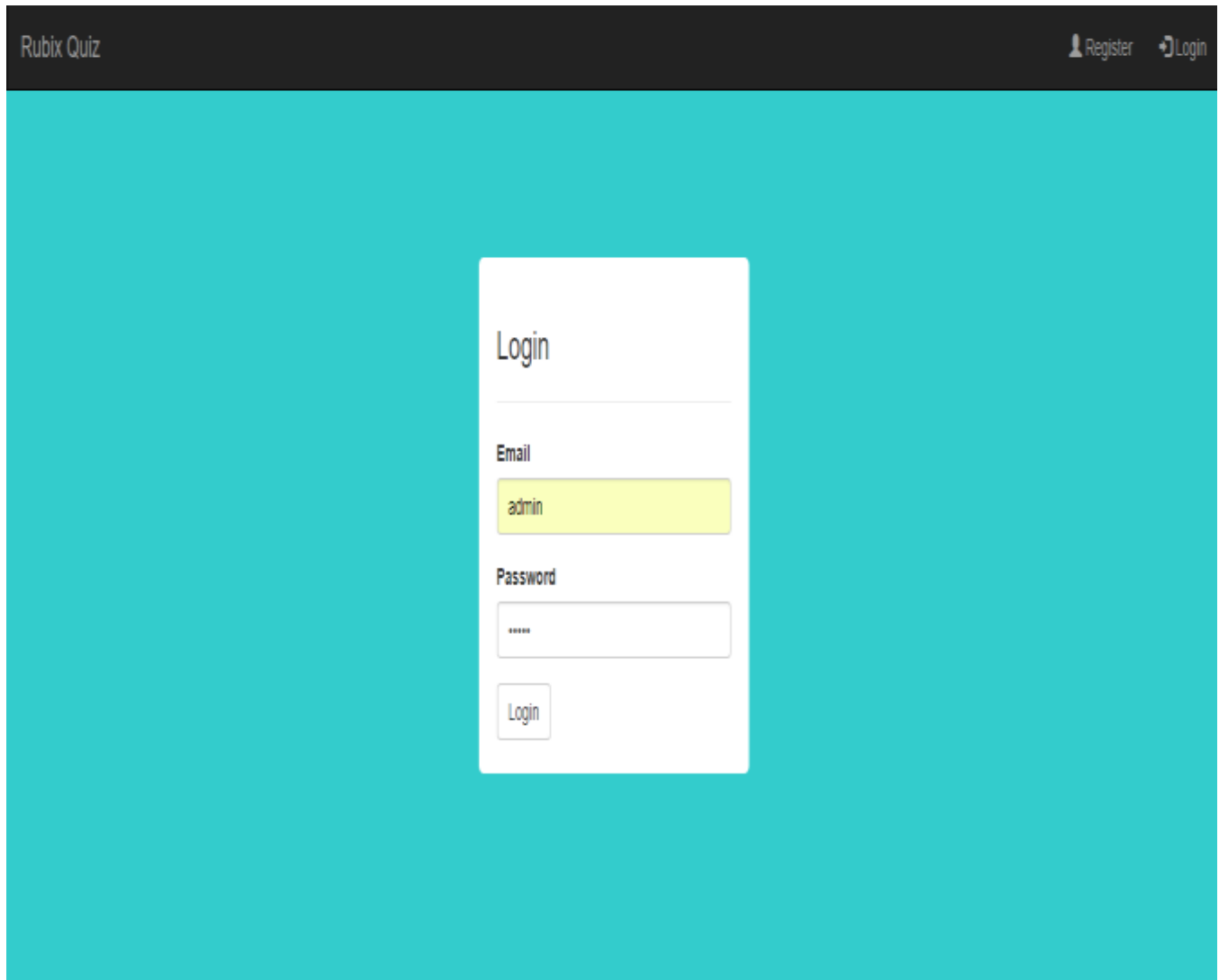
5.3 Implementation:

The first appearance of the RUBIX Quiz is shown below. Here at first sight we can see that there are two parts Login and register. By pressing the login button a user can login either as an admin or a student. Then if someone clicks on to the register button a form will be come out to urge for some information to register such as email id, name, password and confirmation of password are needed.



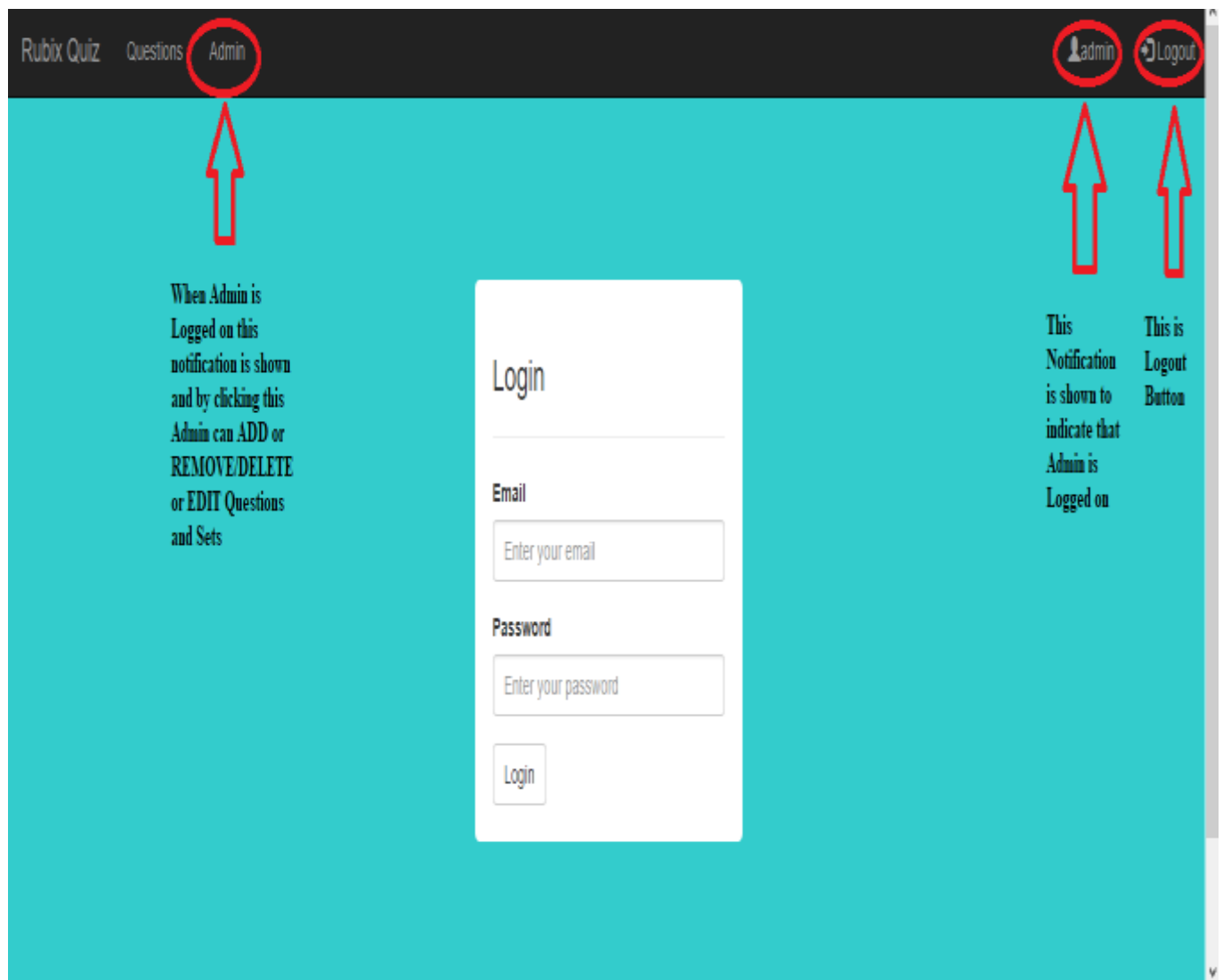
5.2 First Appearance of RUBIX Quiz

At the top of the right of this page we can see that there are two indications one is register and another is logout. When a user logged on the first indication will show user's name. By clicking on logout button the user can be logged out from his or her account.

The image shows a web application interface for 'Rubix Quiz'. At the top, there is a dark header bar with the text 'Rubix Quiz' on the left and two links, 'Register' and 'Login', on the right. The main body of the page has a solid teal background. In the center, there is a white rectangular login form. The form has a title 'Login' at the top. Below the title, there are two input fields: 'Email' and 'Password'. The 'Email' field contains the text 'admin'. Below the 'Password' field, there is a 'Login' button. The 'Email' field is highlighted with a yellow background.

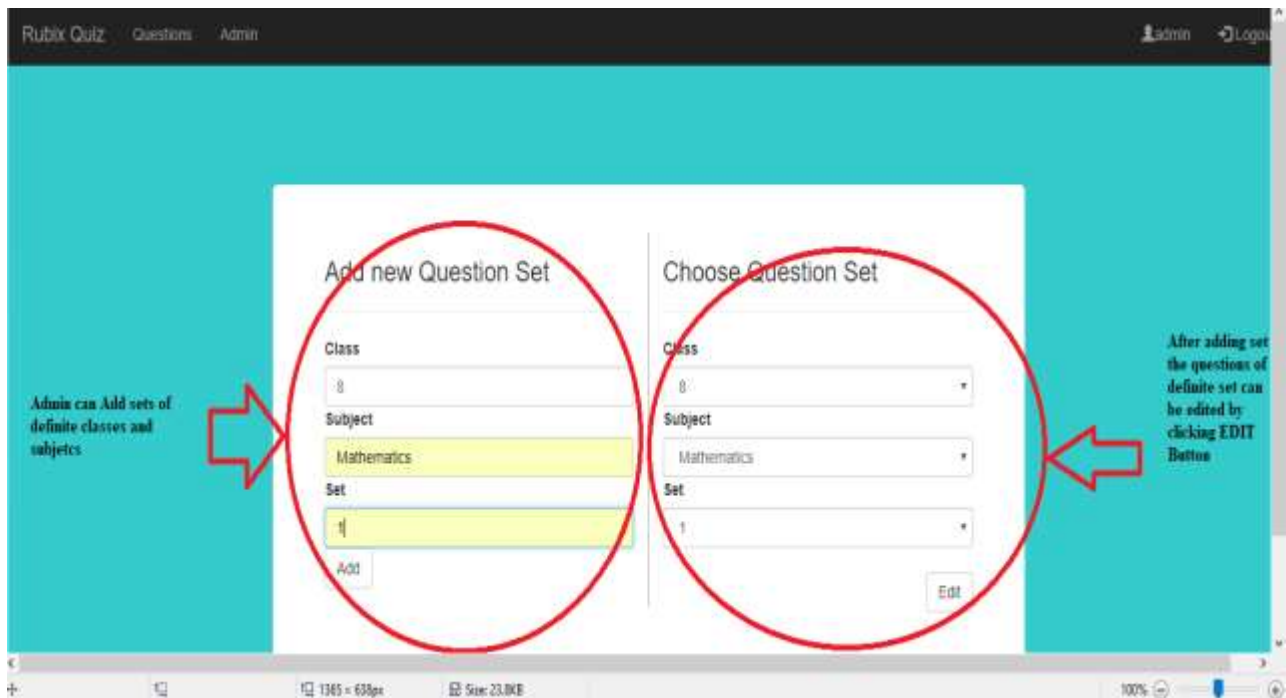
5.3 Admin Login

Admin can Login with own definite password and email id. After Login properly there will be a sign at the top right which indicates who is Logged on.



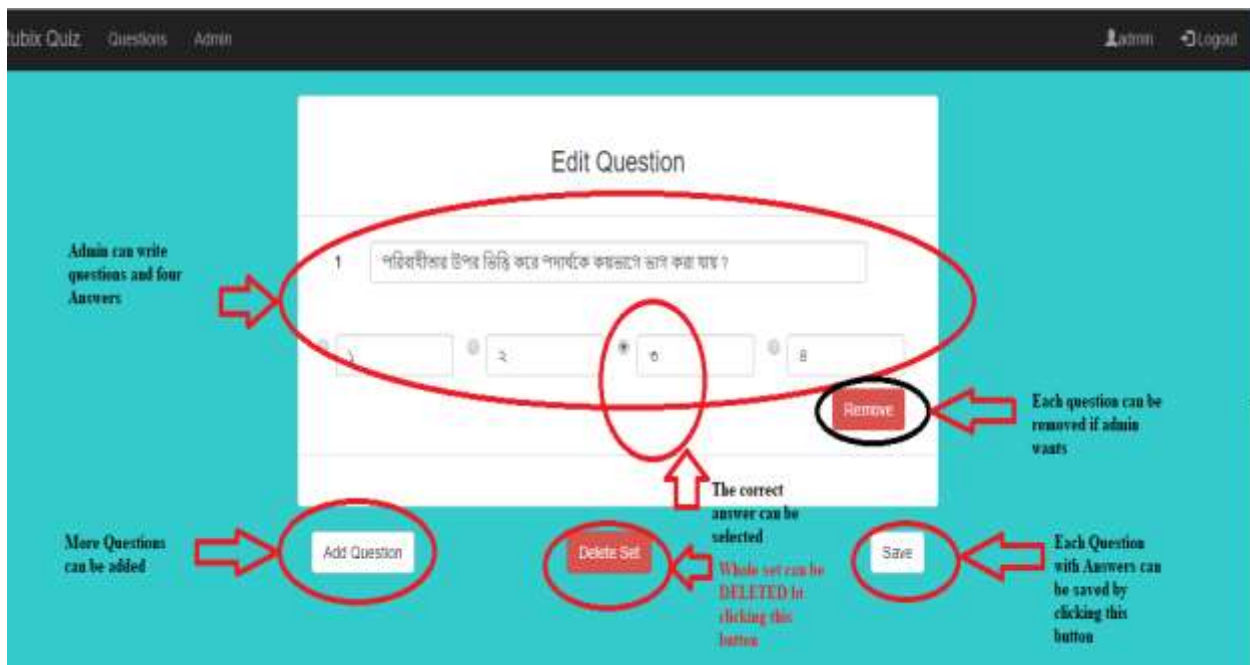
5.4 When the Admin Logged On

When the admin logged on at the left top here indicates a option for admin. By clocking this admin can add classes, subjects, sets and questions with four answers for each question. Moreover admin can delete or edit too of the prescribed orders. Admin only can do this. Here admin symbolizes as the authority. So only admin has the permission to edit delete or add any question classes sets and subjects.



5.5 Admin Works

Here, we can see that admin can add sets of definite classes and subjects. After adding set the questions of definite set can be edited by clicking edit button.



5.6 Editing of Questions and Set

Admin can write questions and four answers. The correct answer can be selected. Each question can be removed if admin wants. Each questions with answers can be saved by clicking save button. However, more questions can be added. Whole set can be deleted by clicking 'delete set' button.

In python there is a huge option that a developer can give any language in database and this site and the database can easily adapted with this. As a result if admin write the questions and answers choices with any other language rather than English like Bangla, Arabian or Chinese, every language will be easily adapted in database with this language means with python. Previously I use PHP to code this language. But PHP did not support any other language than English when I tried to put Bangla on Database for questions with answers. But when I used python it acted totally different than before when I used PHP. It adapted all the questions in other language means Bangla. I filled the database randomly with English or Bangla but it doesn't make any hustle. So I wrote and wrote as much as I need to fill the database with Bangla and English language. Python treated both language in same manner. It didn't differentiate it. As a result I put the two different languages when it was needed to make questions with definite answers like formulas equation or anything regarding to match the student needs.

As a result by trying both programming language like python and PHP I found the python more intelligent more convenient more easy to go and lastly the most smart among all languages to code easily and smartly. So, I just relied on this language to do code. Rather than using the intelligence to adapt all languages python is very east to code. It needs very few lines to code a software or make a site. In RUBIX Quiz I just used avro to write Bangla Font as my keyboard on laptop have no keywords rather than English letter. So I just had to depend on avro keyboard to write the Bangla fonts. When I like to use Bangla font in avro I just had to press F12 button to change the keyboard in Bangla mode. In Bangla mode Bangla language can be written just only write with English alphabets with Bangla pronunciations. Finally the Bangla font just came out with English alphabet and this database easily showed it to the site.

The image shows a web browser window with a dark header bar. On the left of the header is the text "Rubix Quiz". On the right are two links: "Register" with a person icon and "Login" with a key icon. The main content area has a teal background. In the center is a white rectangular form titled "Register". The form contains four input fields: "Name" with placeholder text "Enter your name", "Email" with placeholder text "Enter your email", "Password" with placeholder text "Enter your password", and "Confirm Password" with placeholder text "Confirm password". Below these fields is a "Signup" button.

5.7 New User Register Form

For a new user means student he/she has to be registered in this system. By clicking register button a page will come out to register a user. The form needs name, email address and password and the password has to be confirmed. After filling up the form of the register then the new user has to sign up. When a user click on the sign up button he/she will be fully registered to this site.

Rubix Quiz

Register Login

Login

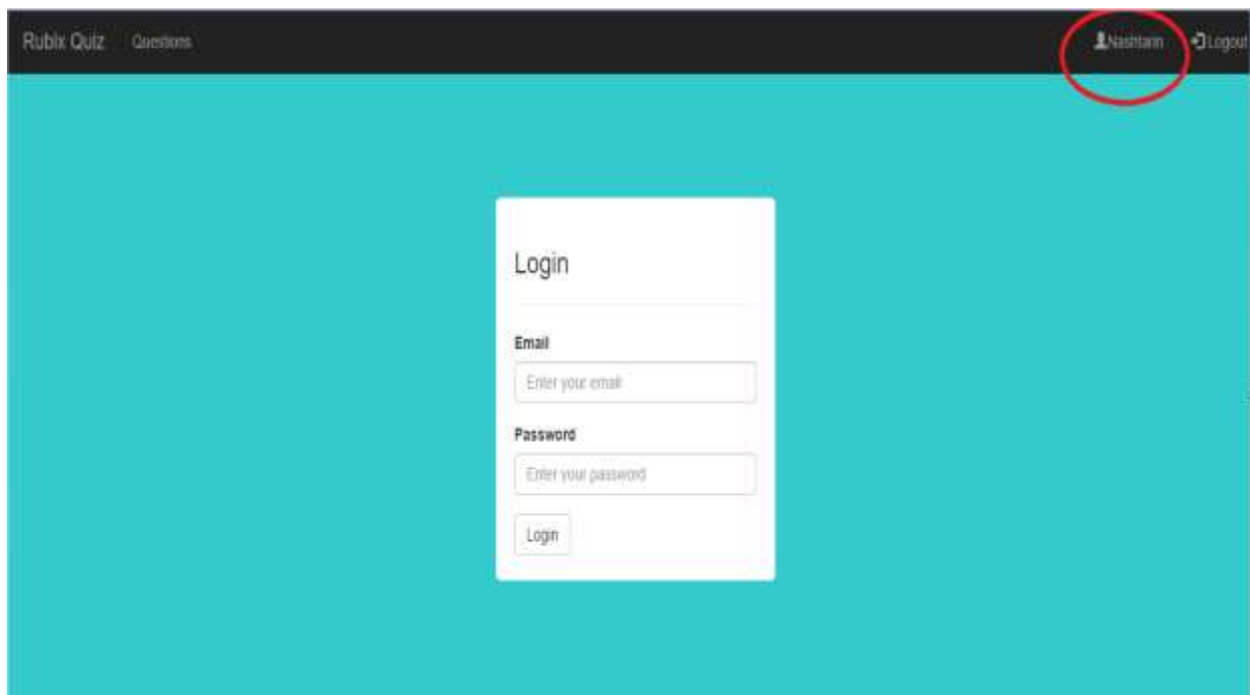
Email

Password

Login

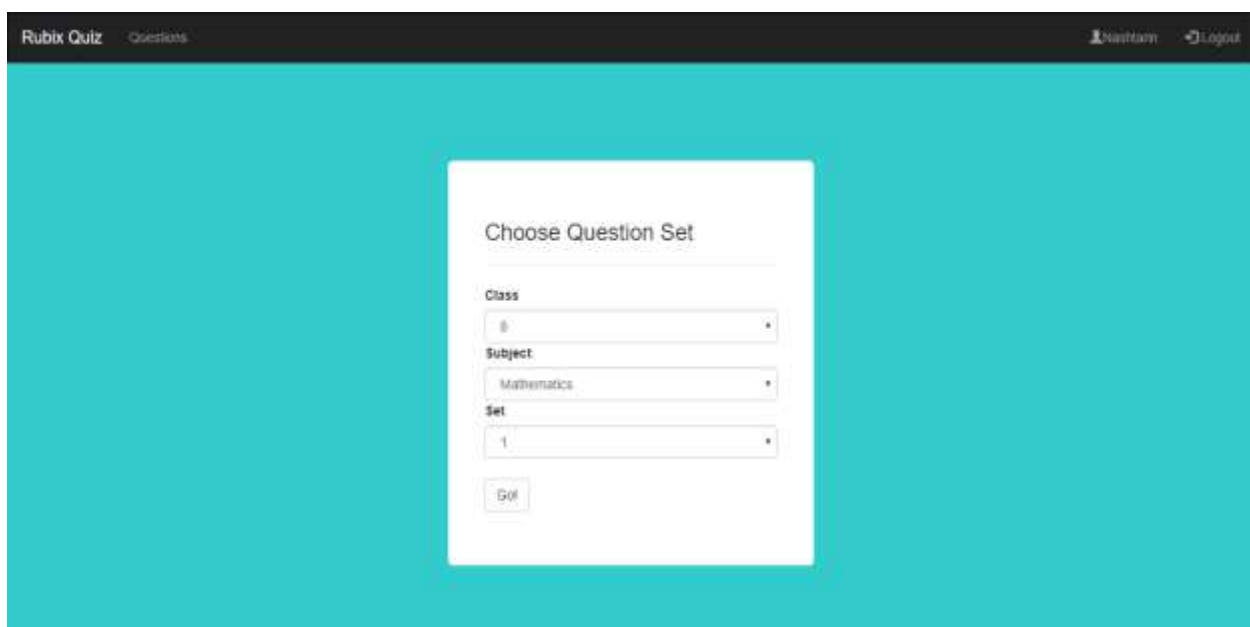
5.8 User Can Login by Giving E-mail and Password

When a user fills up the register form and click the sign up his/her information saved in database. So for login to this site he/she must login. Now for successful login user has to give the definite email and password. When the email and password are given user has to click login button. There the site will start to authenticate with the database. If the site gets the correct authentication it will let the user login to the site successfully.



5.9 Notification of Successful Login

Either admin or user logged on to this site there will be a notification at the top right of the webpage. There will be an indication of the user name. This indication indicates the name of the user and this user is logged on.



5.10 Choice of Set for appearing Examination

After successful login of the user a student can choose subject, class, set of definite subject and class. When student choose class there will come subject and after that there will come set. So after choosing this student have to click go to appear the exam.

Question

1) কোন ব্যক্তি এ টিকা জমা রাখলে তার পরিমার্জক কি বলে?

☐ দুবান্দা ☐ আসল ☐ দুবান্দা-আসল ☐ দুবান্দা বার

2) ৩৯ সংখ্যাটি নিচের কোন সংখ্যার ৬৫%?

☐ ১০০ ☐ ৬৫ ☐ ৬০ ☐ ৩৯

3) ৮% ক্ষতিতে বিক্রয়মূল্য কত টাকা?

☐ ৯২ ☐ ১০০ ☐ ১০৮ ☐ ১০৬

4) বিক্রয়মূল্য ক্রয়মূল্য এর বেড় বাকি বলে লাভ কত?

☐ ৫০ ☐ ১০০ ☐ ১৫০ ☐ ২০০

5.11 The Appearance of Questions

Student has to choose the correct answer among the four answer which he/she things is the best one. A student has to answer all question in a set to check answer and to get the result.

☐ ৫০ ☐ ৬০ ☐ ৬০ ☒ ৬৫

28) ৮% ক্ষতিতে বিক্রয়মূল্য কত টাকা?

☐ ১১২ ☐ ১২৫ ☐ ৯৬ ☒ ৯২

29) ৩৯ সংখ্যাটি নিচের কোন সংখ্যার ৬৫%?

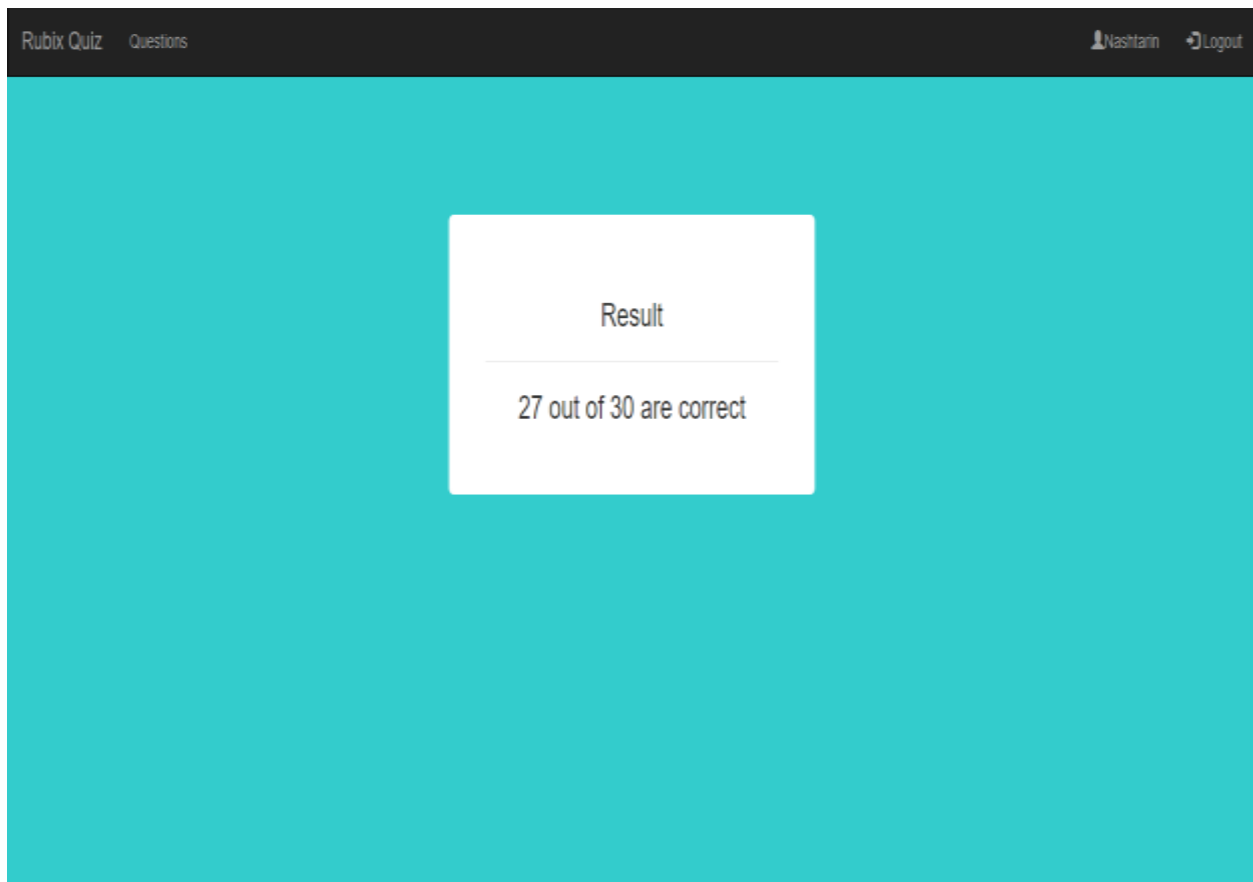
☐ ৫০ ☐ ৬০ ☐ ৬০ ☒ ৬৫

30) ৮% ক্ষতিতে বিক্রয়মূল্য কত টাকা?

☐ ১১২ ☐ ১২৫ ☐ ৯৬ ☒ ৯২

Check Answers

5.12 Checking Answers



5.13 Showing Result

The result will show out and also shows the number of correct answers given out of total number. Hence the result comes out and a student know about his or her preparation.

5.4 Conclusion:

This chapter discusses about the implementation means embodiment of the total project how it works. Moreover, it depicts the login of admin and also how question can be added or removed or deleted. The login of the user is depicted in this chapter. Also it explains how the result has come out.

Chapter 6

Conclusion and Future Work

6.1 Introduction:

In this chapter, it will be discussed about the future work of this site. In future more features will be added like timestamps, explanation of the each wrong answer and the storage of the result of students will be added.

6.2 Selection of Wrong and Correct Answers:

In future new feature will be added that this site can differentiate the wrong answers and correct answers. This will be a pure indication that which questions are answered right and which are answered wrong to the student. So a student can justify themselves properly.

6.3 Explanation for Wrong Answers:

When a student finished exam the result will come out. In future there will be explanations for every wrong questions he/she has given. This will help a lot for these students to prepare for their final examinations.

6.4 Timestamps:

In each set timestamps will be added means for giving a exams a definite time is added. Such that for set A of mathematics of class 8 30 minutes time stamps will be added and this will count down to zero seconds.

6.5 Secure Admin:

Admin will be more secure than before that no one will be get access rather than admin in this site. Moreover, there will be a access more than one admin so than a good administration can be shared.

6.6 Conclusion:

In this chapter it is discussed that how can be this project will be promoted to a very good level. However, this will be a very good site for students from classes six to ten in Bangladesh Academic Curriculum to achieve a good preparation.

References:

- [1] 'Python', [https://en.wikipedia.org/wiki/Python \(programming language\)](https://en.wikipedia.org/wiki/Python_(programming_language)), date:18 January 2018
- [2] 'HTML', <https://en.wikipedia.org/wiki/HTML>, date: 18 January, 2018
- [3]'CSS', [https://en.wikipedia.org/wiki/Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), date: 18 January 2018
- [4] 'Bootstrap', <https://en.wikipedia.org/wiki/Bootstrapping>, date: 18 January 2018
- [5] 'Django', <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>, date: 18 January 2018
- [6] 'Online Quiz System', <https://www.slideshare.net/satyakimitra/online-quiz-system>, date: 18 January 2018

Appendix

Some Simple Source Code of the Project

Codes of Views

```
# -*- coding: utf-8 -*-
from __future__ import unicode_literals

from django.shortcuts import render

# Create your views here.

from django.http import *
from qasite.models import *
from qasite.util import *
from django.shortcuts import *

def index(request):
    page = 'qasite/index.html'
    data = get_data(request)

    data['title'] = "Web Based Quiz System"

    return render(request, page, data)

def question_list(request):
    page = 'qasite/question.html'
    data = get_data(request)
    question_list = []
    for q in Question.objects.all():
        d = {}
        d['question_id'] = q.id
        d['question_text'] = q.question_text
        choice_list = []
        for c in Choice.objects.filter(question_id=q.id):
            choice_list.append(c.choice_text)
        d['choice_list'] = choice_list
        question_list.append(d)
    data['question_list'] = question_list
    data['question_title'] = "All Questions"
```

Codes of Choose Questions

```
{% extends "qasite/base.html" %}
{% block content %}
<style>
    .centered-box{
        margin-top: 100px;
        max-width: 400px;
        padding: 40px;
        background-color: white;
        border-radius: 5px;
    }
</style>

<div class="container centered-box">
    <h3>Choose Question Set</h3>
    <hr>
    <form class="form-group" action="/qasite/answer_question_set/">
        <label for="class_no">Class</label>
        <select class="form-control" name="class_no" id="class_menu"></select>
        <label for="subject_name">Subject</label>
        <select class="form-control" name="subject_name" id="subject_menu"></select>
        <label for="set_no">Set</label>
        <select class="form-control" name="set_no" id="set_menu"></select>
        <br>
        <button class="btn btn-default" type="submit"> Go! </button>
    </form>
</div>
<script>
$(document).ready(function(){
    var class_menu = $('#class_menu');
    var query_string = 'get_class_list/';
    load_menu(class_menu, query_string);
})
$('#class_menu').change(function(){
    var subject_menu = $('#subject_menu');
    class_no = this.value;

```

Codes of Admin Edit Questions

```
{% extends "qasite/base.html" %}
{% block content %}
<style>
    .centered-box{
        margin-top: 100px;
        max-width: 400px;
        padding: 40px;
        background-color: white;
        border-radius: 5px;
    }
</style>

<div class="container centered-box">
    <h3>Choose Question Set</h3>
    <hr>
    <form class="form-group" action="/qasite/answer_question_set/">
        <label for="class_no">Class</label>
        <select class="form-control" name="class_no" id="class_menu"></select>
        <label for="subject_name">Subject</label>
        <select class="form-control" name="subject_name" id="subject_menu"></select>
        <label for="set_no">Set</label>
        <select class="form-control" name="set_no" id="set_menu"></select>
        <br>
        <button class="btn btn-default" type="submit"> Go! </button>
    </form>
</div>
<script>
$(document).ready(function(){
    var class_menu = $('#class_menu');
    var query_string = 'get_class_list/';
    load_menu(class_menu, query_string);
})
$('#class_menu').change(function(){
    var subject_menu = $('#subject_menu');
    class_no = this.value;

```

Codes of answer question set

```
/style>



## Codes of Models.py



```
class Question(models.Model):
 question_text = models.CharField(max_length=5000)

class Choice(models.Model):
 question_id = models.ForeignKey(Question)
 choice_text = models.CharField(max_length=5000)

class Correct(models.Model):
 choice_id = models.ForeignKey(Choice)

class Set(models.Model):
 question_id = models.ForeignKey(Question);
 set_no = models.IntegerField()

class Subject(models.Model):
 set_id = models.ForeignKey(Set)
 subject_name = models.CharField(max_length=5000)

class Class(models.Model):
 subject_id = models.ForeignKey(Subject)
 class_no = models.IntegerField()

class User(models.Model):
 email = models.CharField(max_length=2000)
 password = models.CharField(max_length=2000)
 name = models.CharField(max_length=2000)

class Admin(models.Model):
 user_id = models.ForeignKey(User)
```



42


```

Codes of HTML of Admin edit

```
<div class="container centered-box">
  <div class="col-md-6 vertical-right-divider">
    <h3>Add new Question Set</h3>
    <br>
    <form class="form-group" action="/qasite/admin/add_question/">
      <label for="class_no">Class</label>
      <input class="form-control" name="class_no">
      <label for="subject_name">Subject</label>
      <input class="form-control" name="subject_name">
      <label for="set_no">Set</label>
      <input class="form-control" name="set_no">
      <input type="hidden" name="choice_1" value="">
      <input type="hidden" name="choice_2" value="">
      <input type="hidden" name="choice_3" value="">
      <input type="hidden" name="choice_4" value="">
      <input type="hidden" name="correct_choice" value="">
      <button class="btn btn-default" type="submit"> Add </button>
    </form>
  </div>
  <div class="col-md-6">
    <h3>Threes / Martin Cat / h3</h3>
  </div>
</div>
```

The End