

PROJECT REPORT
ON
Online Quiz Software
(EasyToQuiz)

B.Tech (CE) Sem-IV
In the Subject of
Software Engineering Practices (CE-IV)

Darshan Parmar (CE-085) (19CEUOS050)

Dhyey Patel (CE-092)(19CEUOS145)

Divyesh Patel (CE-094) (19CEUXS020)

Under the Guidance of
Prof. Jigar Pandya
Prof. Brijesh Bhatt
Prof. Pinkal Chauhan



Department Of Computer Engineering
Faculty of Technology,
Dharmsinh Desai University, Nadiad.

DHARMSINH DESAI UNIVERSITY

College Road, NADIAD-387001(Gujarat)



CERTIFICATE

This is to certify that the term work carried out in the subject of
Software Engineering Practice and recorded in this report is
Bonafide work of

Mr. Darshan Parmar (Roll No.: CE085, ID: 19CEUOS050),

Mr. Dhyey Patel (Roll No.: CE092, ID: 19CEUOS145)

And

Mr Divyesh Patel (Roll No.: CE094, ID: 19CEUXS020)

Of **B.Tech Semester 4th** in the branch of Computer Engineering during the
academic year 2020-21.

Prof. Brijesh S. Bhatt
(Project Guide and Associate Professor)
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Dr. C.K. Bhensdadia
Head of CE Dept.,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Index

1. Abstract	4
2. Introduction.....	5
3. Software Requirement Specification.....	7
4. Design.....	11
5. Implementation Detail.....	19
6. Layouts.....	26
7. Conclusion.....	32
8. Limitation and Future Extension.....	33
9. Bibliography.....	34

1. Abstract

- EasyToQuiz is an online Quiz generating software in which user can create Questionnaire (A Teacher or a Test taker creates a set of questions with a choice of answers for a group of students, devised for the purposes of collecting data and taking tests) easily. Further user can give their response on it and the quiz owner can see all responses on that Quiz with their points.
- Users:-Host (test-taker), students(those who are giving test).

2. Introduction

2.1 Introduction of Online Quiz Software

In this world of internet most of the education related work is done on online platform whether it is assignment submitting, learning different things online, programming and most important is taking exams for evaluation of learners. So the task for taking MCQ exams easily is the motive of our software.

2.2 Features of EasyToQuiz

- Users can create their account by signup.
- Users (test-taker) can create new quiz containing Questions with their Option for other users (students) and can also set answer if needed for particular quiz.
- Users (students) can give exam online and submit it.
- User (quiz-owner) can see all responses with marks for particular students for particular quiz which he/she has created and consider it for evolution.

2.3 Tools/Technologies Used

- Technologies:**

- Django
- Python
- HTML5
- CSS 3
- MySQL
- Bootstrap
- JavaScript
- jQuery

- Tools**

- Git
- Visual Studio Code
- Photoshop

- Platform**

- Apache-Xampplocalhost(Server)

3. Software Requirement Specifications

● Functional Requirements:-

R 1. Login/Sign-Up and Handling accounts:

R 1.1 Login and Sign-Up:

Input: A button to select login or Sign-Up

Output: A new window containing details of login/Sign-Up.

Description: Users who are giving/hosting tests can sign in for the first time and further he/she can login through username and password. For Sign-Up user needs to enter an email ID, their username and a password for this website. If email-ID or username is already present in the database then a message triggers showing a message that "*Username or Email already exist*" And when users want to login they need to enter only username and correct password. If credentials are wrong then message triggers showing "*Invalid username or password*".

R 1.2 Handle Sign-Up information:

Description: We are handling a database containing data of the user's login and their account. It contains their name, Email-ID, password and some other information. Whenever new user sign-up then information of that user will be added to the database.

R 2. Creating Questionnaire:

R 2.1 Create new Quiz:

Input: a button for create quiz

Output: a new webpage containing structure to create quiz.

R 2.2 Add Title:

Input: A text box which accept the title for the Quiz

Output: Heading will be added to Quiz.

Description: Heading may contain short description of Quiz like

R 2.3 Creating Questions:

R 2.3.1 Add Question:

Input: Add Question button

Output: new question will be added at the end

Description: Whenever admin clicks on add question a new section will be added containing all functionalities related to the question.

R 2.3.2 Question description:

Description: Each section of question consist of a text area Where admin write question and a button to add new option and a button for delete this Question.

R 2.3.3 Add option and delete option:

Description: with every section of question it consist add option button on click a new text box appears where host can write the option, beside that option a delete button over there on click it that option will be deleted.

R 2.3.4 Delete Question :

Description: There is a delete button with every section of Question which deletes that entire question.

R 2.4 Save Quiz:

R 2.4.1 Handle Quiz data:

Input: A button to save quiz to database.

Output: Quiz will be added to database.

Description: By clicking save quiz button quiz is saved to database, afterwards quiz-code is generated of 6 alphabets and a message showing “Quiz created successfully” and button for set the answer will appear on screen.

R 2.4.2 Set answers:

Input: A button to set answer after submitting quiz.

Output: Created quiz opens and user have to select correct option to set it as answer key.

Description: By selecting all correct options submit it as answer key by clicking submit button (this answer key response is used for further evolution of other user who are giving this quiz.)

R 3. Sending Quiz-code:

Description: After creating quiz, software will generate quiz code of 6 characters automatically and send the quiz information along with quiz-code to that user via email.

R 4. Give Quiz:

R 4.1 Start Quiz:

Input: Quiz-code

Output: Quiz of that quiz-code will be shown on screen with containing title, description and question.

Description: user need to enter the quiz code to enter into the quiz, if quiz-code is wrong then a message will shown contains “*invalid Quiz-code!*”

R 4.2 Fill Quiz-form:

Input: Quiz information

Output: quiz data will be added to response table

Description: user can select appropriate options for the given questions and click on submit to send their response. If response have sent successfully then a message will shown that “*Response have been Recorded*”.

R 5. Handling Responses and Statistics:

R 5.1 Handling Responses:

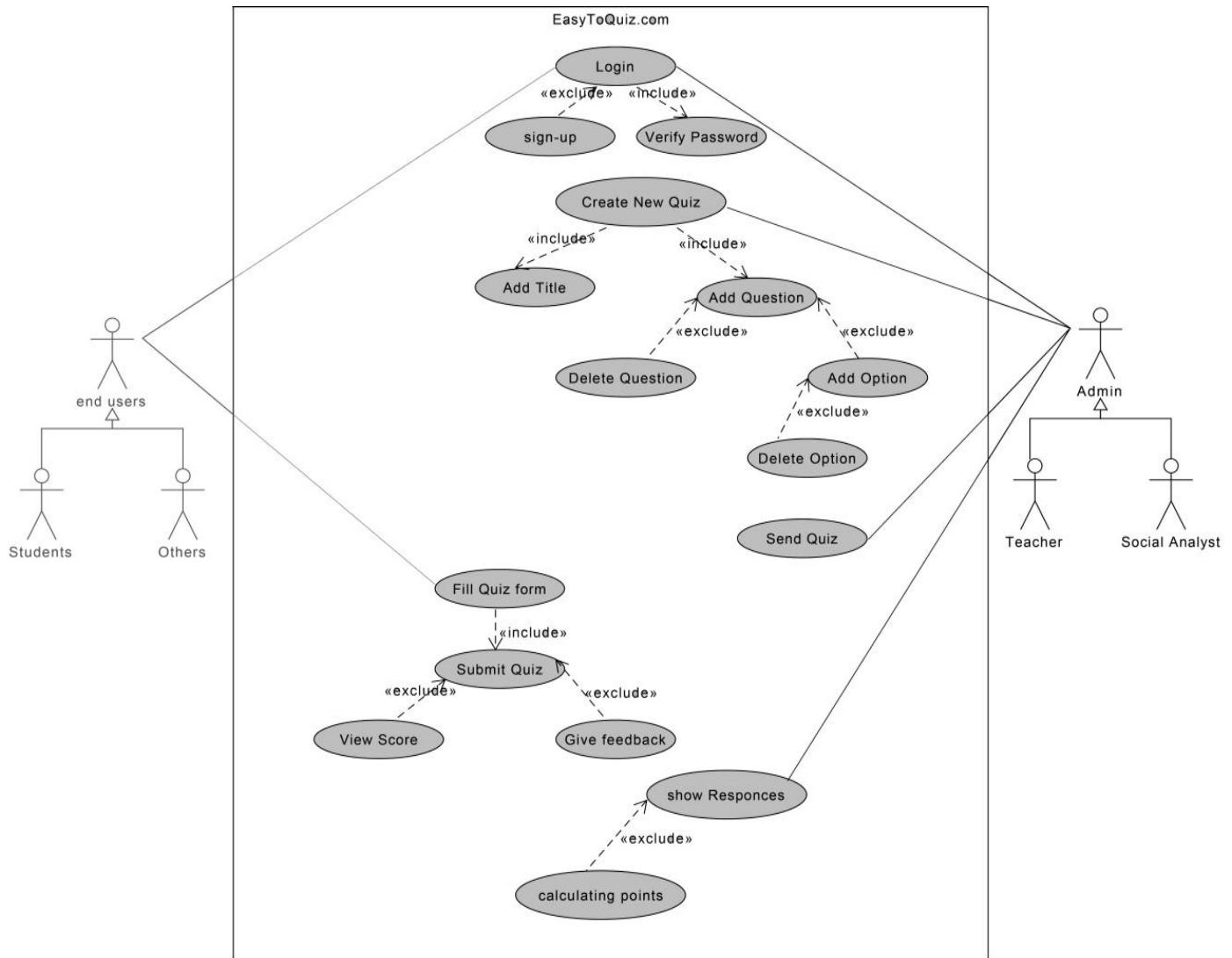
Description: The submitted answers by a particular end-user will be stored in database and can be seen by the host in the response-section.

R 5.2 Calculating point:

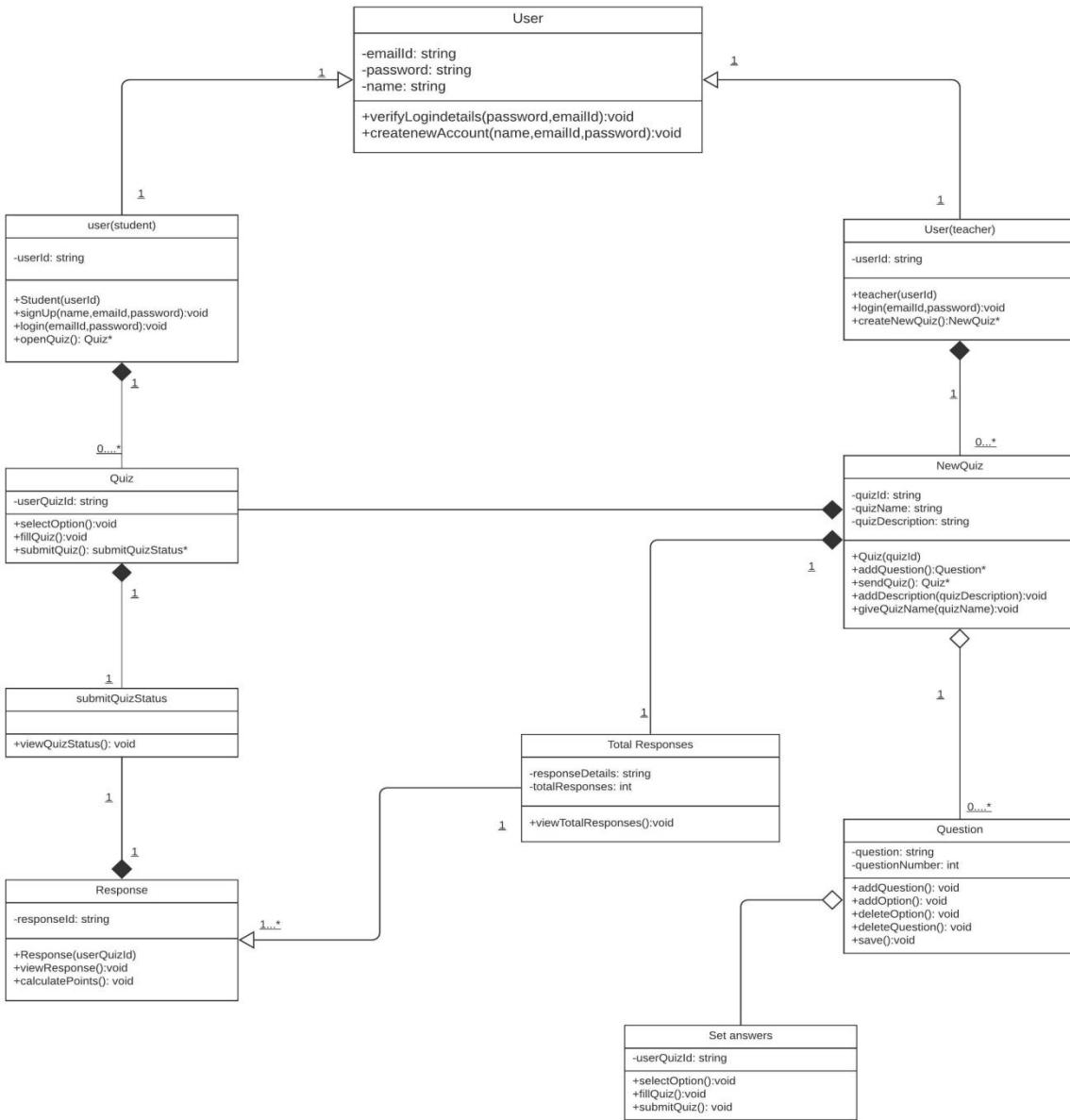
Description: All responses from users will stored in database, and it can be shown by the owner of the quiz. Quiz-owner can also see the total correct answers of a particular response out of total questions and also see the correct answer (which provided by user him/her-self) and answer of that student.

4. Designs:

4.1 Use-case Diagram:

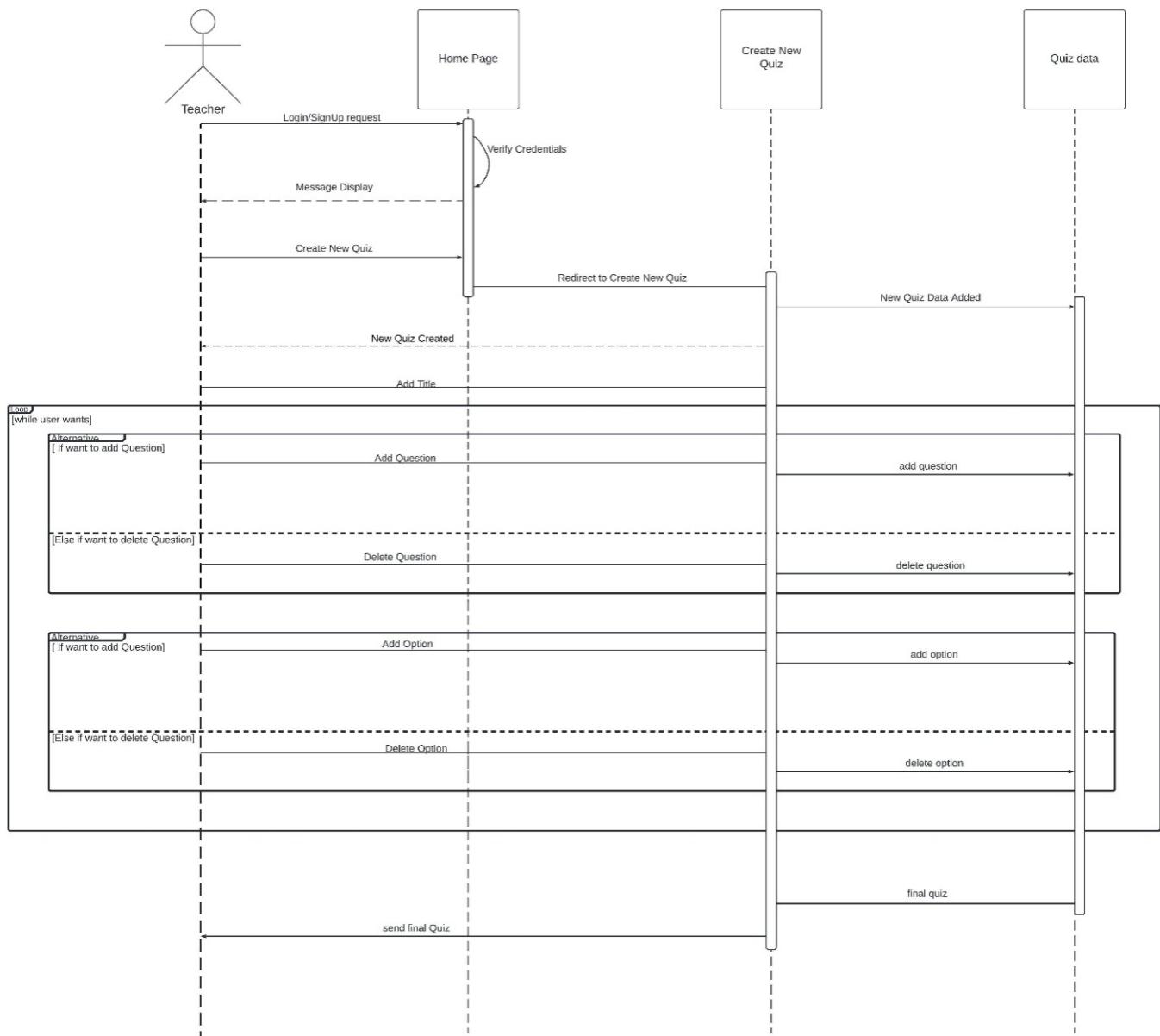


4.2 Class Diagram:

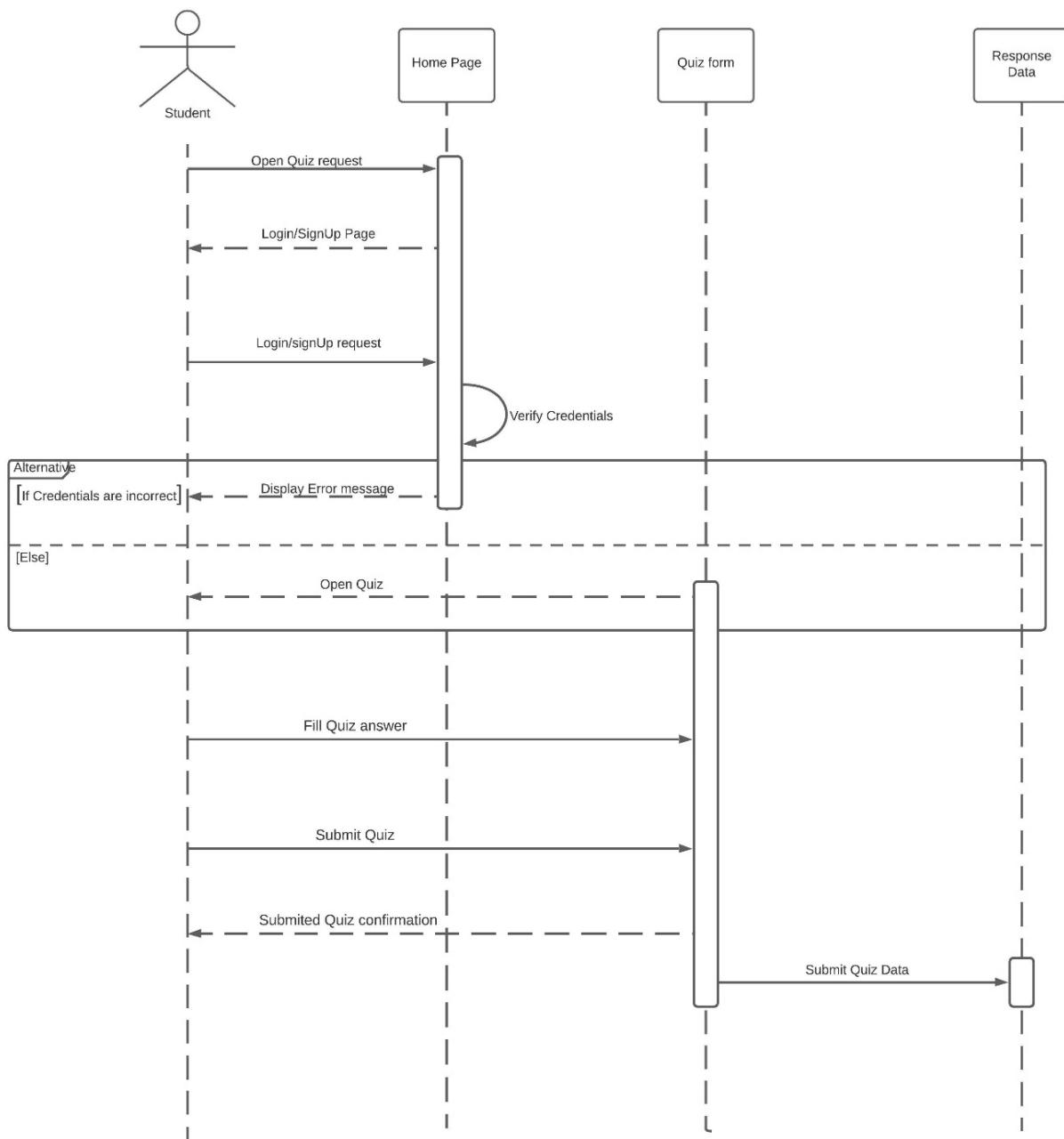


4.3 Sequence Diagram:

4.3.1 Sequence Diagram-I

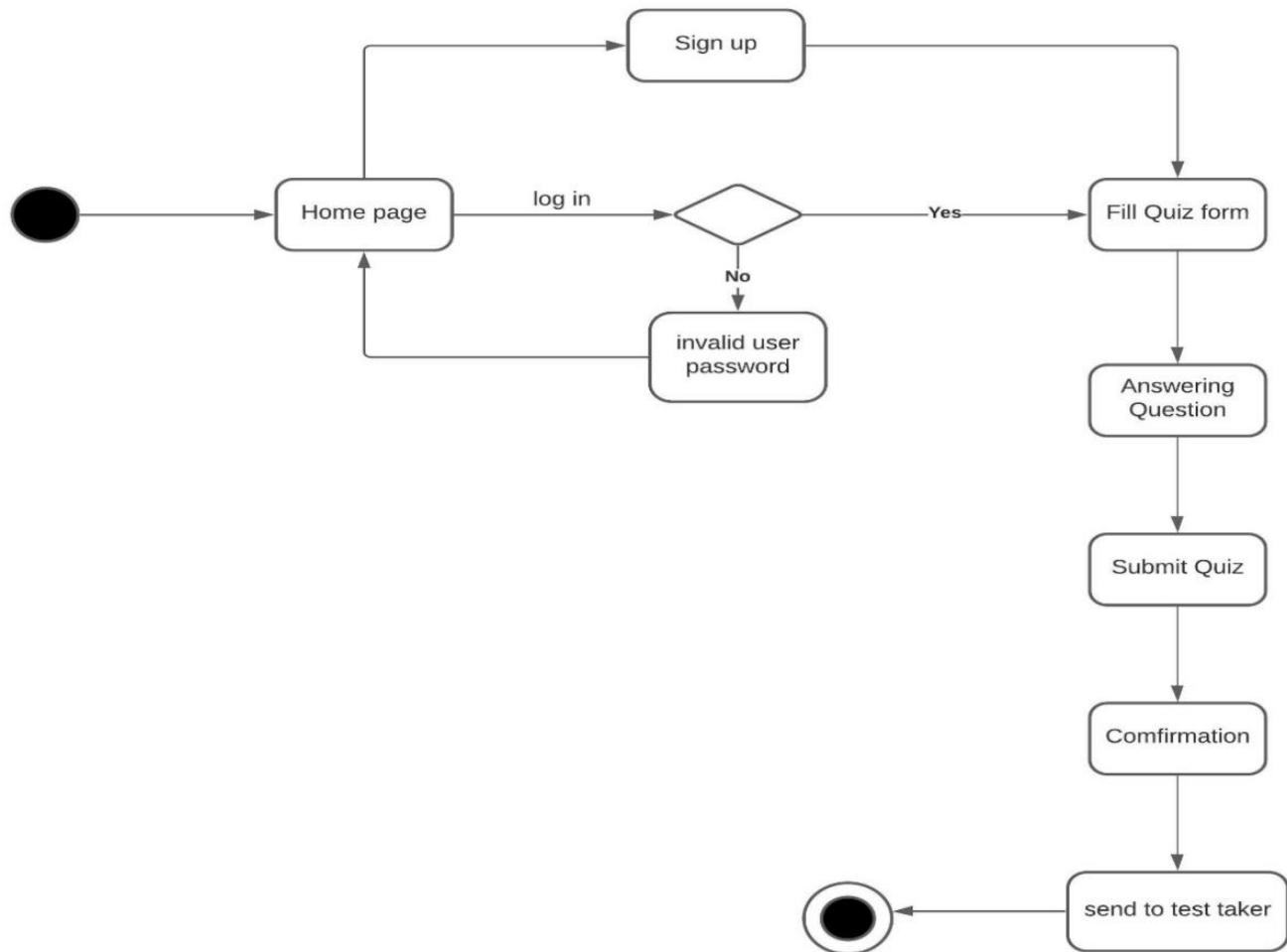


4.3.2 Sequence Diagram-II

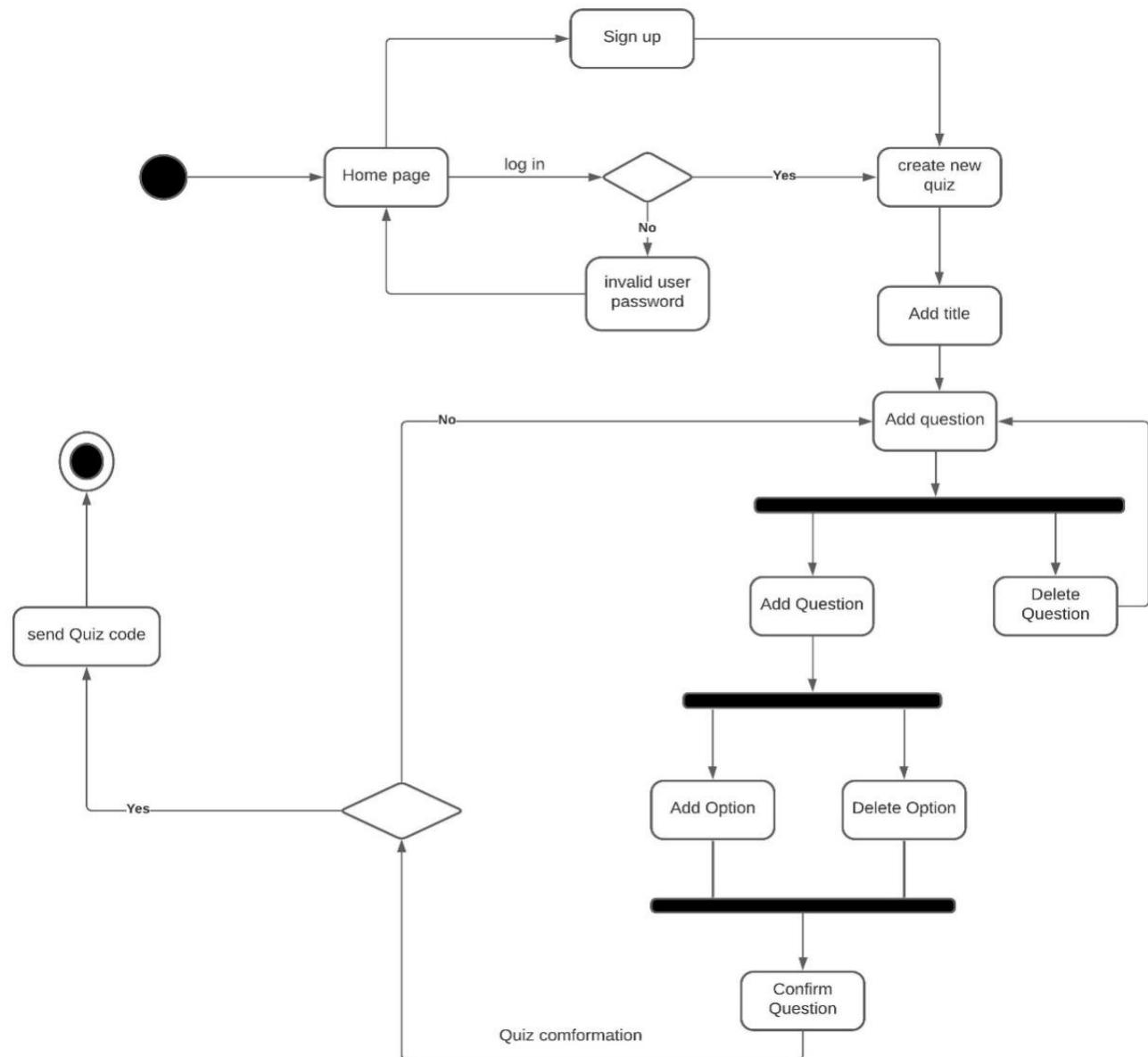


4.4 Activity Diagrams :

4.4.1 Activity Diagram-I

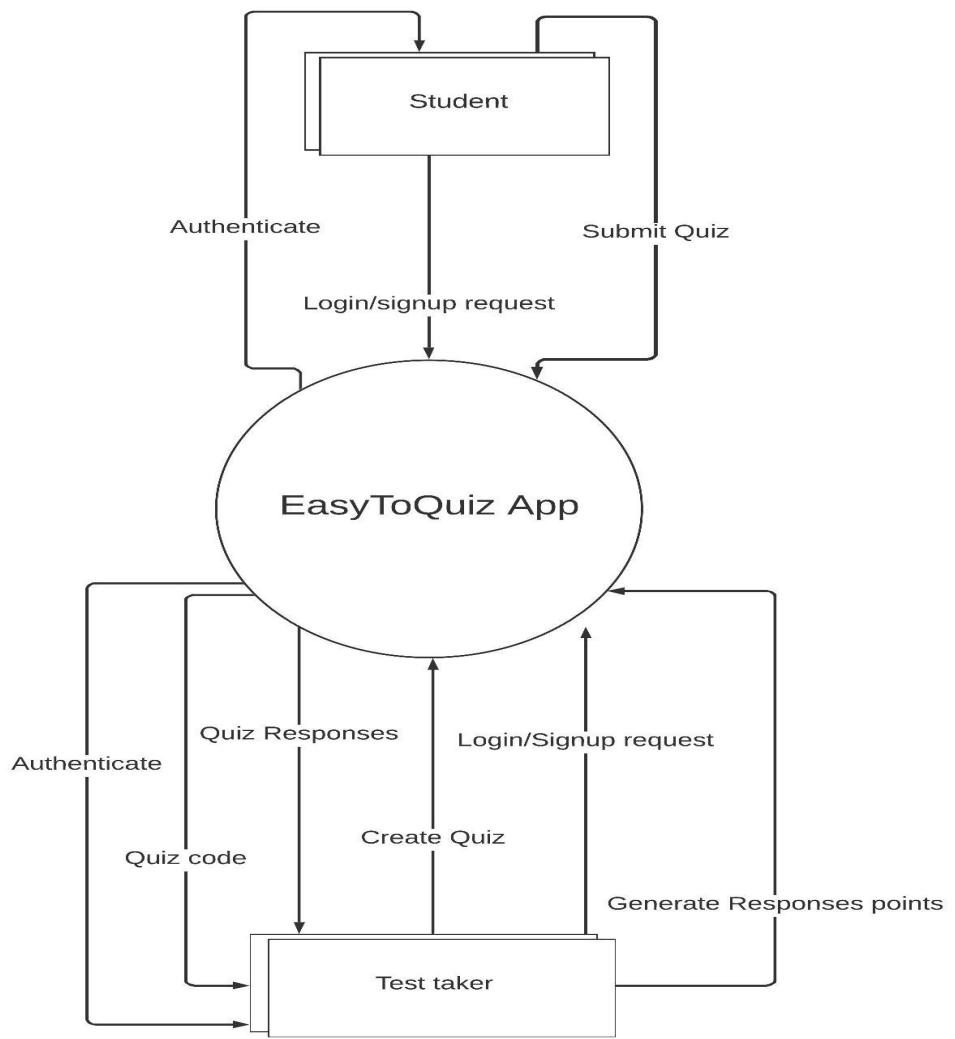


4.4.2 Activity Diagram-II

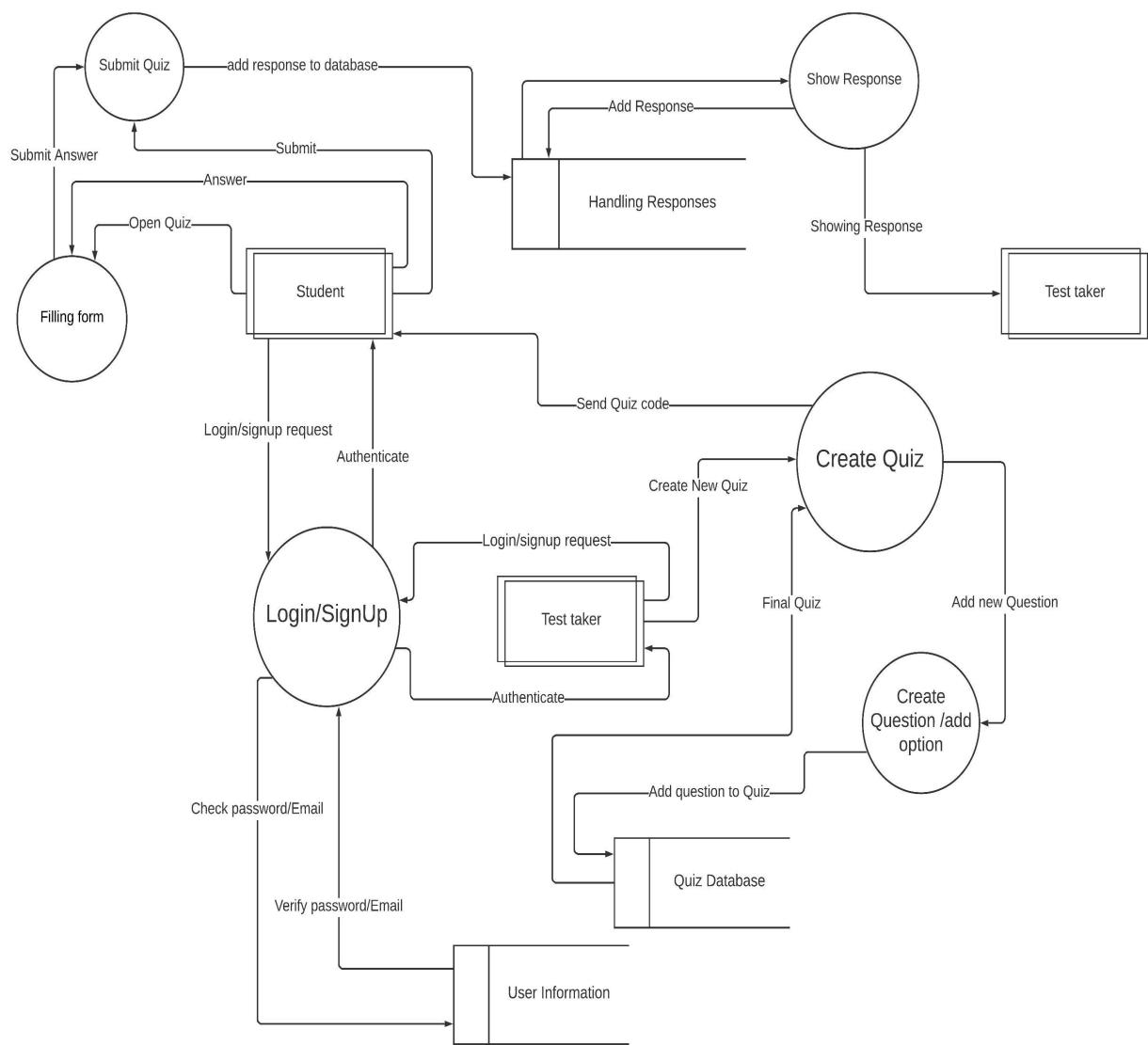


4.5 DFD Diagram :

4.5.1 DFD LEVEL-0



4.5.2 DFD LEVEL-1



5 Implementation Details

5.1 Modules:

5.1.1 User Registration and Login:

Information of new user is taken and stored to the users' database. Whenever that user will login then last login will be updated to the table.

- **Login View:**

```
def login(request):
    if request.method == 'POST':
        sname = request.POST.get('username', '')
        spass = request.POST.get('password', '')
        user = auth.authenticate(username=sname, password=spass)
        if user is not None:
            auth.login(request, user)
            return HttpResponseRedirect('/EasyToQuiz')
        else:
            messages.error(request, "Username or password invalid")
            return render(request, 'login.html')
    else:
        return render(request, "login.html")
```

- Sign-Up view:

```
def userregistration(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('/EasyToQuiz/')
    if request.method == 'POST':
        sname = request.POST.get('username', '')
        semail = request.POST.get('email', '')
        spass = request.POST.get('pass', '')
        spass1 = request.POST.get('cpass', '')
        if spass==spass1:
            if User.objects.filter(username=sname).exists():
                messages.error(request,"username already exist.")
                return render(request, "userregistration.html")
            elif User.objects.filter(email=semail).exists():
                messages.error(request,"This email is already connected to an account.")
                return render(request, "userregistration.html")
            else:
                messages.success(request,"Register Successfull, please Login again for confirmation.")
                s = User.objects.create_user(username = sname, email=semail, password=spass)
                subject = 'Welcome to EasyToQuiz'
                message = 'Thank you '+sname+',\nHave a nice journey on EasyToQuiz!'
                email_from = settings.EMAIL_HOST_USER
                recipient_list = [semail,]
                send_mail( subject, message, email_from, recipient_list )
                s.save()
                return HttpResponseRedirect('/EasyToQuiz/login')
        else:
            messages.error(request,"confirm password is not same as password! please check!")
            return render(request, "userregistration.html")
    else:
        return render(request, "userregistration.html")
```

5.1.2 Create Quiz:

Users are allowed to make questionnaires with their own requirements. This module includes major functional requirements like addQuestion, addOption, deleteOption, deleteQuestion and some more. When a user makes a new questionnaire and submit it then it will stored into the database which is handling by multiple tables like quiz data, question data and option data.

- Saving Quiz View:

```
def savingquiz(request):
    if request.method == 'POST':quizid=0;quiz_id=""
        while 1:
            quiz_id=generate_quizid()
            exist = quiz_data.objects.raw("SELECT * from userlogin_quiz_data WHERE quizid=%s", [quiz_id])
            if len(exist)==0:break
            quiztitle=request.POST.get('title','');quizdescription=request.POST.get('description','')
            mail=request.POST.get('smail','')
            u_id =int(request.POST.get('u_id',''))
            q = quiz_data(quizid=quiz_id ,quiztitle=quiztitle , description=quizdescription ,username_id=u_id);q.save()
            array2=request.POST.get('array2','').split(",");array=request.POST.get('array','').split(",")
            for i in range(0,int(request.POST.get('x',''))):
                if array2[i+1]=='1':
                    questiontitle=request.POST.get('QuestionTitle-'+str(i+1),'');questiontype=False
                    quizid=q.id
                    Q = Question_data(qtitle=questiontitle,qtype=questiontype,quizid_id=quizid)
                    Q.save()
                    for j in range(1,int(array[i+1])+1):
                        option = request.POST.get("option-"+str(i+1)+"-"+str(j))
                        if option:
                            questionid=Q.id
                            op = option_data(option=option,questionid_id=questionid,quizid_id=quizid);op.save()
            subject = str(quiztitle)
            message = 'You have successfully created Quiz on EasyToQuiz!\nQuiz information is as follow:\nQuiz Title: '+str(quiztitle) + '\nQuizID: '+str(quizid)
            email_from = settings.EMAIL_HOST_USER
            recipient_list = [mail,];send_mail( subject, message, email_from, recipient_list )
            context = {"QuizID" : q.quizid }
            return render(request, "quiznext.html",context)
        else:
            return HttpResponseRedirect['/EasyToQuiz']
```

5.1.3 Give Quiz/Add Response:

Users are also allowed to answer a particular quiz. Whenever a user submits their answers, software will add it to database. It is also handled by multiple tables like response data and answers.

- Give Quiz view:

```
def quiz(request):
    if request.method == 'POST':
        user_id=request.POST.get('user_id','')
        quizid=request.POST.get('quizcode','')
        x = quiz_data.objects.raw("SELECT * from userlogin_quiz_data WHERE quizid=%s", [quizid])
        if len(x)==1:

            exist = response_data.objects.raw("SELECT * from userlogin_response_data WHERE userid=%s and quizid_id=%s", [user_id, x[0].id])
            if len(exist)!=0:
                return render(request, "submitquiz.html")
            elif x[0].response_status:
                Q_data=Question_data()
                for y in x:
                    title=y.quiztitle
                    description=y.description
                questions = Question_data.objects.raw("SELECT * from userlogin_question_data WHERE quizid_id=%s", [x[0].id])
                option = option_data.objects.raw("SELECT * from userlogin_option_data WHERE quizid_id=%s", [x[0].id])
                context = {"questions":questions , "options":option, "title":title, "description": description}
                return render(request, "quiz.html",context)
            else:
                context = [{"mymessage":"This quiz is no longer accepting response, try to contact owner of the Quiz!!"}]
                return render(request, "quizdata.html",context)
        else:
            context = {"mymessage":"No Quiz available for this ID" }
            return render(request, "quizdata.html",context)
    else:
        return HttpResponseRedirect('/EasyToQuiz')
```

- **View-Response view:**

```

def view_response(request):
    username=request.POST.get("username","")
    user1=User.objects.raw("SELECT * from auth_user WHERE username=%s",[username])
    quizid=(request.POST.get("quizid",""))
    quiz=quiz_data.objects.raw("SELECT * from userlogin_quiz_data WHERE id=%s", [quizid])
    correct_answer_list=[];userid=quiz[0].username_id
    correct_response=response_data.objects.raw("SELECT * from userlogin_response_data WHERE quizid_id=%s and userid_id=%s",[quizid,userid])
    for cr in correct_response:
        correct_option=option_data.objects.raw("SELECT * from userlogin_option_data Where id=%s",[cr.answer_id])
        correct_answer_list.append(correct_option[0].option)
    responses=response_data.objects.raw("SELECT * from userlogin_response_data WHERE quizid_id=%s and userid_id=%s",[quizid,user1[0].id])
    question_list=answer_list=[]
    for response in responses:
        ques=Question_data.objects.raw("SELECT * from userlogin_question_data WHERE id=%s",[response.questionid_id])
        question_list.append(ques[0])
        answer=option_data.objects.raw("SELECT * from userlogin_option_data WHERE id=%s",[response.answer_id])
        answer_list.append(answer[0])
    is_answered=True;i=0
    if len(correct_response)!=0:
        is_answered=True
        response_list=zip(question_list,answer_list,correct_answer_list)
        for question,answer,correct_answer in response_list:
            if answer.option==correct_answer:
                i+=1
        response_list=zip(question_list,answer_list,correct_answer_list)
    else:
        is_answered=False
        response_list=zip(question_list,answer_list)
    x=len(question_list)
    context={"response_list":response_list , "quiz":quiz[0], "total":x , "correct":i,"name":username,"email":user1[0].email, "answered":is_answered}
    return render(request,"view_response.html",context)

```

5.2 Data dictionary:

5.2.1 Users

Name of Field	Datatype	Length
ID	Int	11
Password	Varchar	128
Username	Varchar	150
Email	Varchar	254

5.2.2 Quiz_data

Name of Field	DataType	Length
ID	Int	11
Quiz Title	Varchar	40
Description	Varchar	1000
Username_id	Int	11
Quizid	Varchar	40
Response_Status	Boolean	1

5.2.3 Question_data

Name Of Field	DataType	Length
ID	Int	11
Q_title	Varchar	500
Q_type	Boolean	1
Quizid_id	Int	11

5.2.4 Option_data

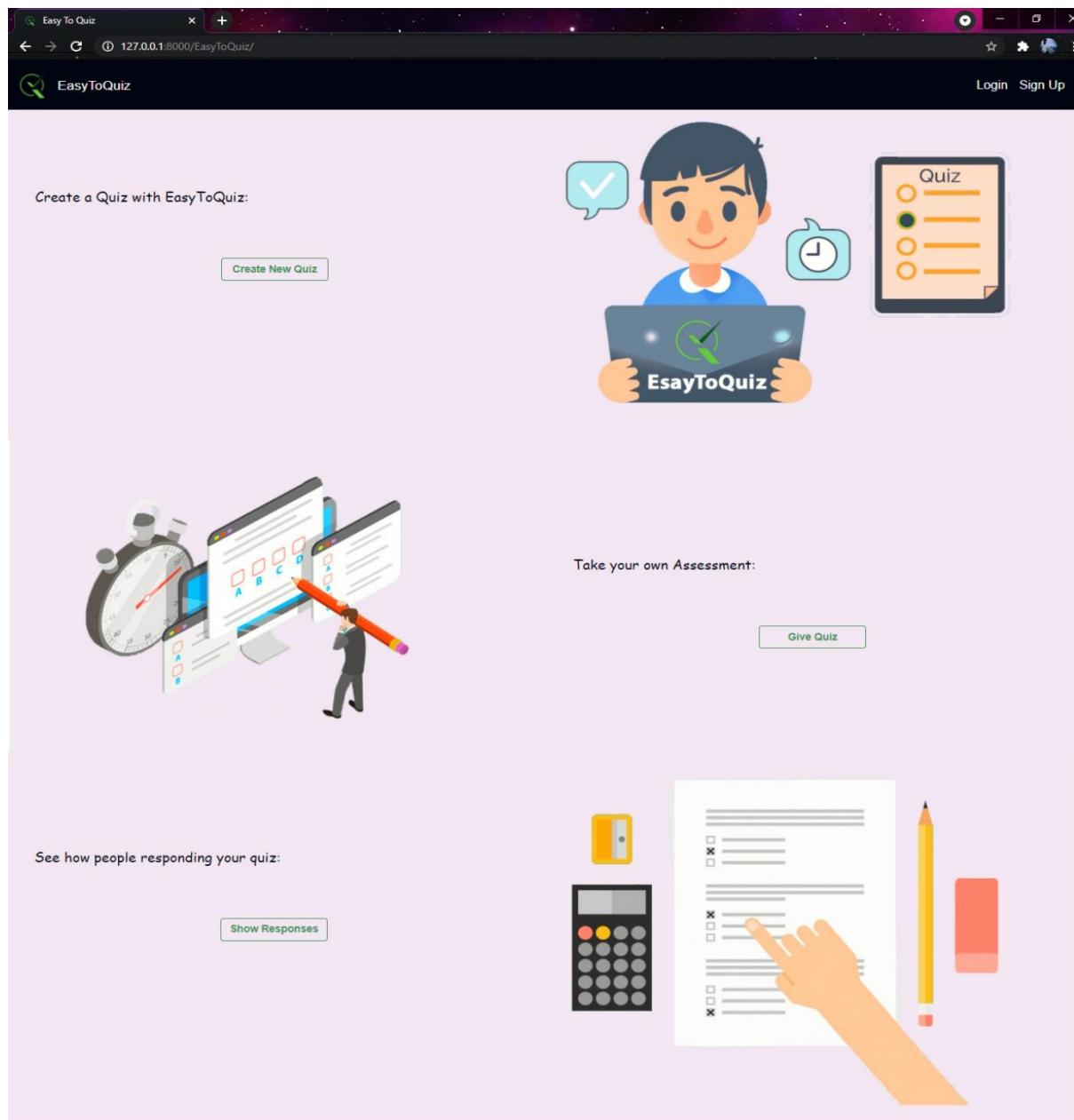
Name Of Field	DataType	Length
ID	Int	11
Option	Varchar	500
Questionid_id	Int	11
Quizid_id	Int	11

5.2.5 Response_data

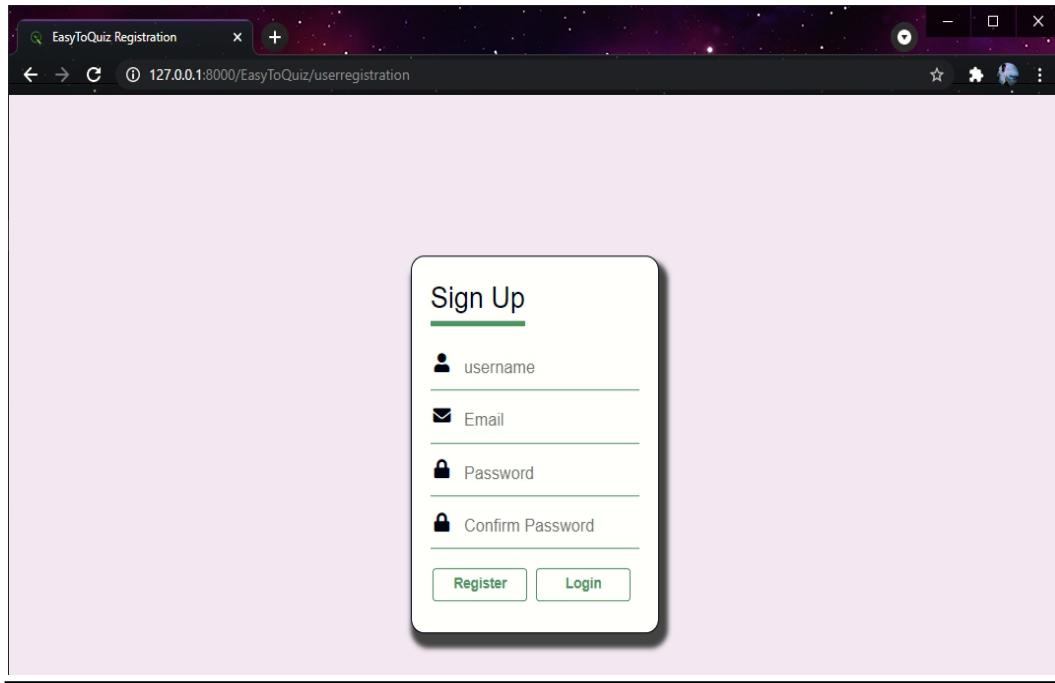
Name Of Field	DataType	Length
ID	Int	11
Answer_id	Int	11
Questionid_id	Int	11
Quizid_id	Int	11
Userid_id	Int	11

6 Layouts

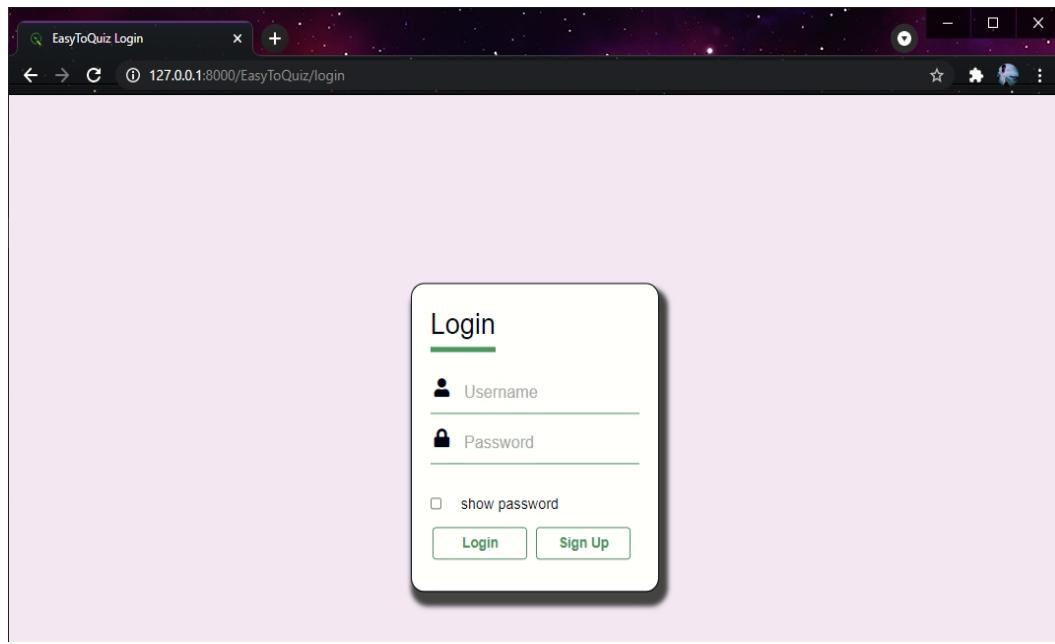
6.1 Home:



6.2 Sign-up:



6.3 Login:



6.4 Create Quiz:

The screenshot shows a web browser window titled "Create New Quiz" at the URL "127.0.0.1:8000/EasyToQuiz/createquiz". The user is logged in as "DARSHAN". The page displays a "New Quiz" form with a question being added. The question title is "Maths" and the description is "All the best!". The question text is "what is value of PI?", with a correct answer of "3.14" and a green checkmark icon. A large green plus sign icon is visible on the left side of the question area.

The screenshot shows a web browser window titled "Quiz Created" at the URL "127.0.0.1:8000/EasyToQuiz/quiznext2". The user is logged in as "DARSHAN". The page displays a message "Your Quiz ID is: 6E5418" with a copy icon, and two buttons: "Set Answer" and "Back to home".

6.5 Give Quiz:

The image consists of two screenshots of a web-based quiz application.

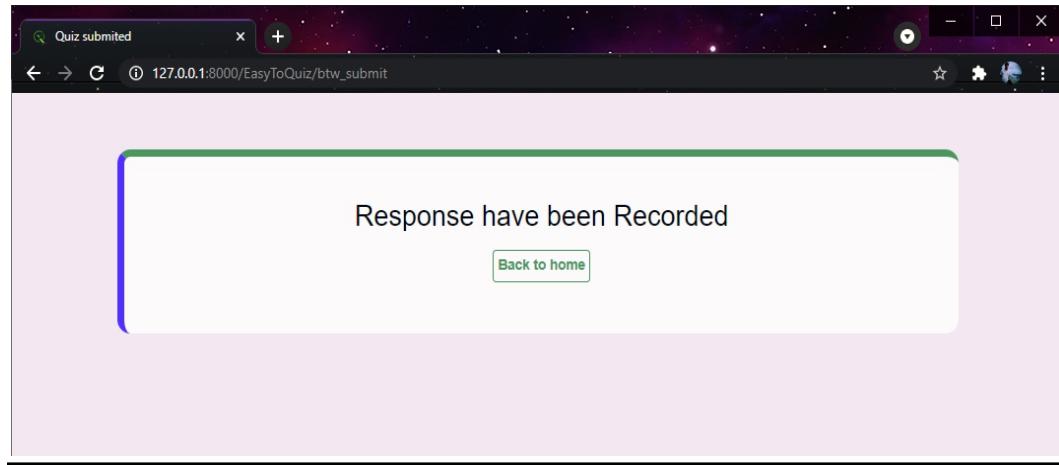
Screenshot 1: Quiz Code

A modal window titled "Quiz Code" is displayed. It contains a text input field with the value "6E5418". Below the input field are two buttons: "start Quiz" (in green) and "Back" (in white).

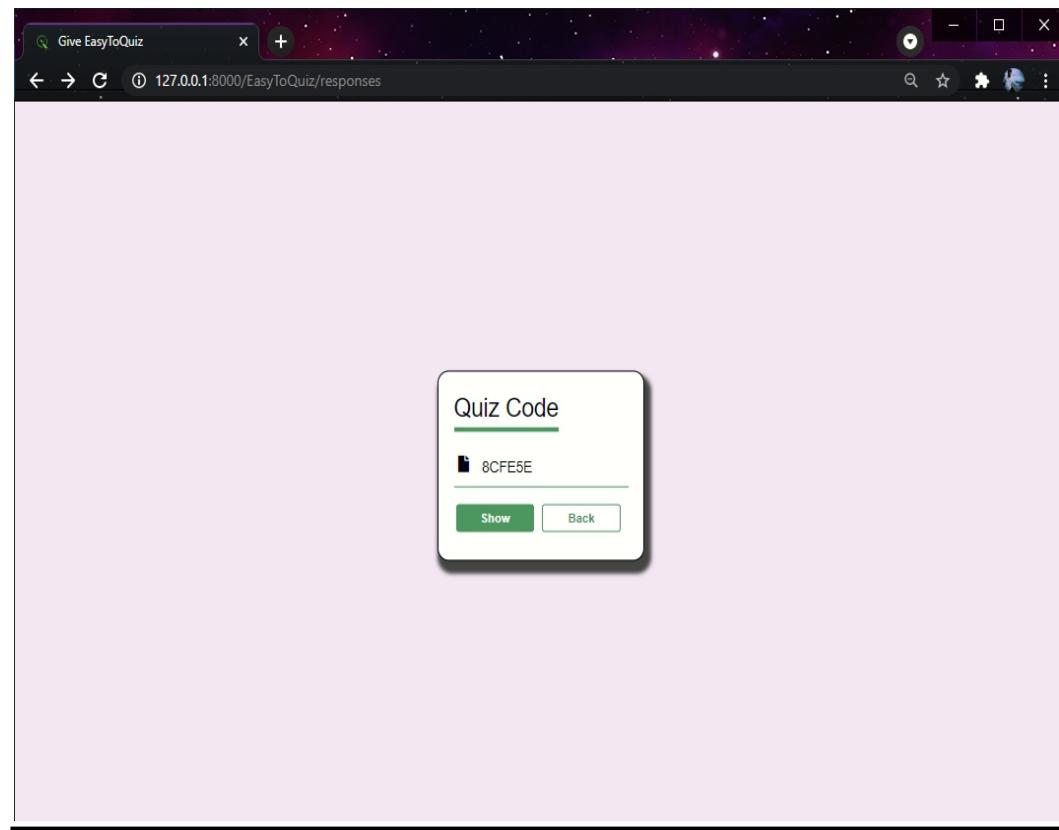
Screenshot 2: Quiz

The main page is titled "Quiz" and shows a user "Divyesh2" logged in. The page displays the following content:

- Maths**
you have 1 hour for this exam. all the best. don't refresh your page during exam. total marks(40)
- what is value of PI?**
 - 3.14
 - 2.14
 - 7/22
 - 22/3.14
- what is value of e?**
 - 2.73
 - 2.2
 - none of the above
- what is log₂(base 2)?**
 - 1



6.6 Show Responses:



Easy To Quiz

127.0.0.1:8000/EasyToQuiz/responses_data

EasyToQuiz

darshan2.0 Logout

Click the name below to see Full response.

DARSHAN

Dhyey

Divyesh

Darshan2

Dhyey2

Divyesh2

Response Status

This screenshot shows a list of user names on a web page. Each name is enclosed in a purple-bordered box. At the bottom right of the list is a 'Response Status' section with a blue toggle switch.

View Response

127.0.0.1:8000/EasyToQuiz/view_response

EasyToQuiz

darshan2.0 Logout

Maths

Name : Divyesh2
Email : divyesh2@gmail.com
Score : 2/4

what is value of PI?

Marked Answer: 3.14
Correct Answer: 3.14

what is value of e?

Marked Answer: 2.2
Correct Answer: 2.73

what is log2(base 2)?

This screenshot shows a 'View Response' page for a user named 'Divyesh2'. It displays a 'Maths' section with the user's name, email, and score. Below this are three questions with their respective marked and correct answers.

7 Conclusion

- The main purpose of our project is to develop an application that offers new aspects of learning and improving knowledge in educational area. Most of the available applications are entertainment based, which mostly do not contribute to the academic enhancement of the students.
- The theme of our application is to provide user to practice for any objective test conducted at online level, so in this application we mainly focused on both create and attend quiz. This application is very helpful for teachers to make their own quiz for their students for get results faster and accurate.
- We have learned a lot about web based application and learn how front end and backend programming is important for a software like this. We have found that the development process is hard and time consuming, but it can be managed by our team work. We hope that other developers will take advantage from our development.

8 Limitation and future extension

8.1 Limitation:

- Type of Question is only single correct option.
- Quiz link generation is not implemented only quiz-code (6 letters) is generated.
- Cannot change account password once generated.
- Cannot delete quiz or user account once generated.
- Once quiz created then it cannot be modified further.
- Multiple responses are not allowed for any quiz if needed.

8.2 Future Extension:

- We will work on our limitations and try to improve our application.
- We will try to make it more user-friendly and add few more functionalities.
- Can implement time factor for better real effect (not necessary as user can stop accepting response to any quiz).

9 Bibliography

9.1 Websites:

- ⇒ <https://stackoverflow.com/>
- ⇒ <https://www.djangoproject.com/>
- ⇒ <https://www.geeksforgeeks.org/>
- ⇒ <https://www.w3schools.com/>

9.2 Search Engine:

- ⇒ www.google.com