Our first Web Application: A Quiz Portal!

Hi there! Thank you so much for checking out our Quiz Portal. This is the final project of the IT Workshop-2 course in our first year at IIIT Hyderabad! Here’s a little something about the functionalities implemented and troubles faced.

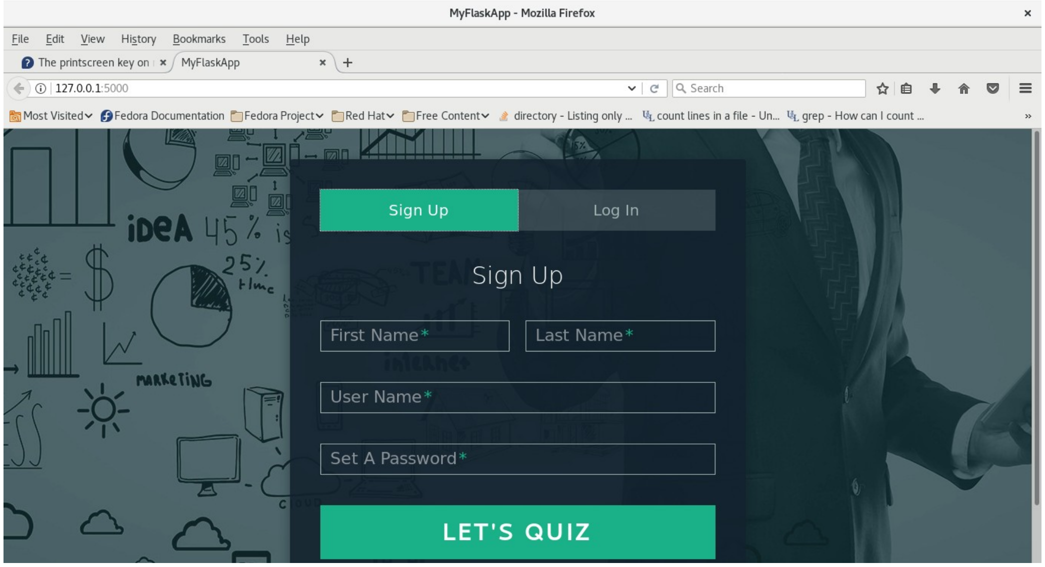
So this web app is written with the help of Python, Flask and SqlAlchemy. You can check out the README file for more information as to how to install the dependencies. It is based on the basic Model-View-Controller model. Model means the database for which we used Sql Alchemy, the view means the user interface(html, css etc) and the controller: main logic of our code and how it works(Flask and Python).

1) Login and register:

The most basic and the first thing anyone does while creating a web app: allowing the user to create his or her account so that only he/she can access it. So we basically created a table in our database called ‘User’ and kept storing the login data of all users in it.

```
class User(Base):
    __tablename__ = "users"

    id = Column(Integer, primary_key=True)
    username = Column(String)
    password = Column(String)
    staterestore = Column(Integer)
    idleft=Column(Integer)
    name = Column(String)
```

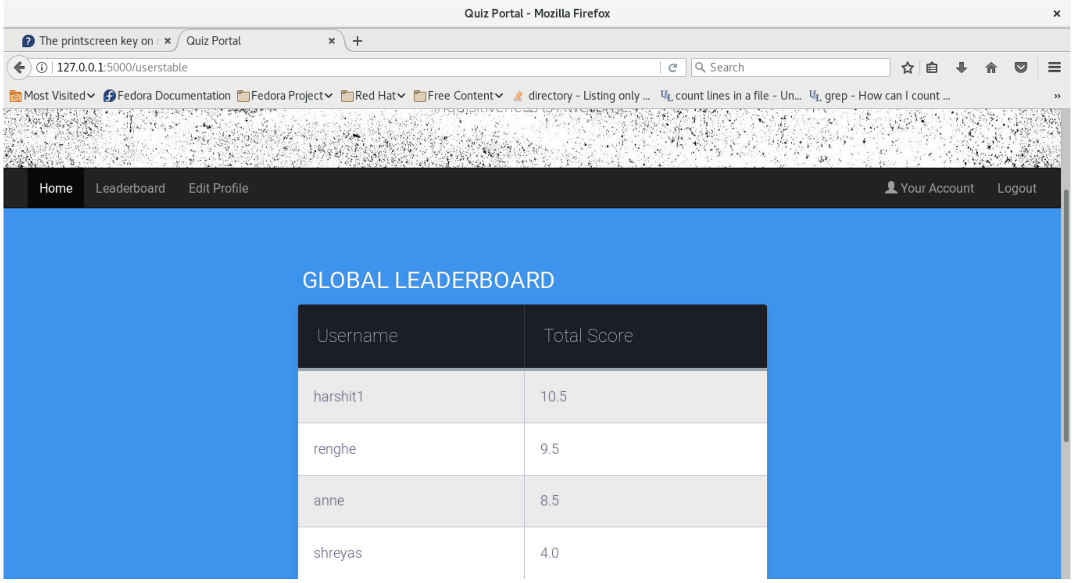
So this is a table that stores the username and password and some other attributes(which we shall see later). Whenever a user tries to login, we look into this table and check if the password entered matches with the one in the database.



2) The Home Page

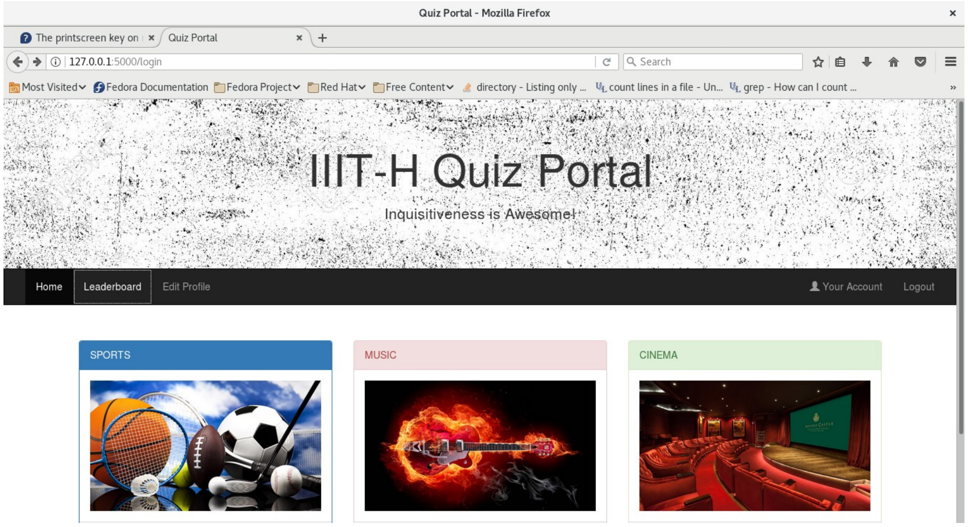
This is where we offer three categories of Quizzes: Sports, Music and Cinema. You can click on any of these to see the subcategory quizzes and take them. The navbar has three options: Leaderboard, Edit profile and Your Account. The leader board is like a global leaderboard: It displays the total score of all users in all the quizzes in descending order.

The Your Account page show your statistics, that is, whatever quizzes you have taken, your highest score in those and number of times you have attempted it.



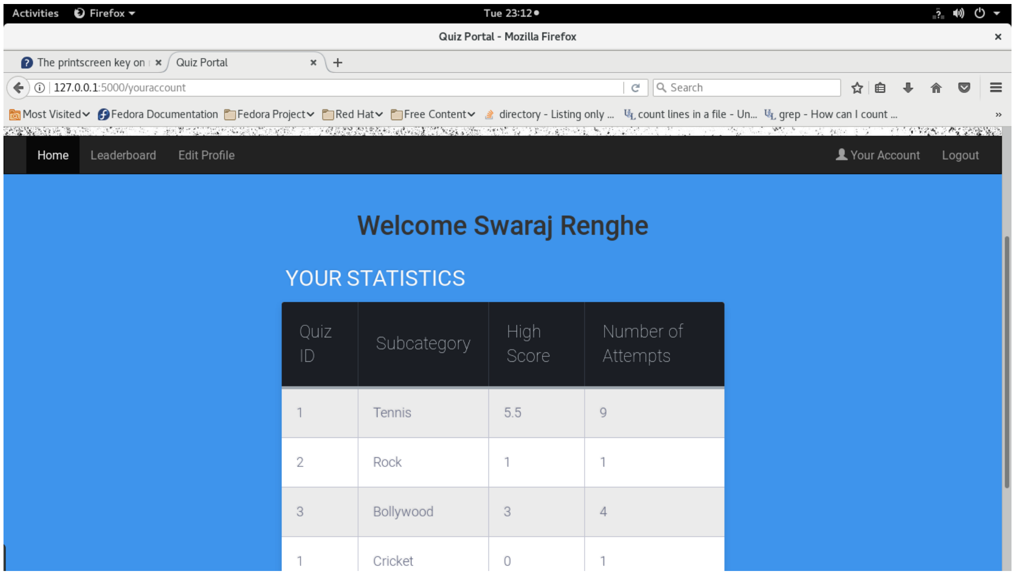
GLOBAL LEADERBOARD

Username	Total Score
harshit1	10.5
renghe	9.5
anne	8.5
shreyas	4.0



IIIT-H Quiz Portal

Inquisitiveness is Awesome!



Welcome Swaraj Renghe

YOUR STATISTICS

Quiz ID	Subcategory	High Score	Number of Attempts
1	Tennis	5.5	9
2	Rock	1	1
3	Bollywood	3	4
1	Cricket	0	1

For implementing the global leaderboard:

```
listofusers=s.query(User).all()

    leaderboard = []

    for i in listofusers:
        finalscore=0
        scoresofuser = Scores.query.filter_by(username=i.username).all()
        for j in scoresofuser:
            finalscore+=j.high_score

        x=(i.username, finalscore)
        if(i.username!="admin"):
            leaderboard.append(x)
```

So essentially, we fetch the names of all users and for each user, we check the Scores table and add up the user’s score in each quiz. The scores table is another table in our database which keeps track of the quizzes that you attempt and your highest score in them.

```
class Scores(db.Model):
    __tablename__ = 'scores'

    id=db.Column(db.Integer, primary_key=True, autoincrement=True)

    username = db.Column(db.String(200))
    quiz_id = db.Column(db.Integer)
    subcategory = db.Column(db.String(200))
    score_uptilnow = db.Column(db.Integer)
    attempts = db.Column(db.Integer)
    high_score = db.Column(db.Integer)

    question_left = db.Column(db.Integer)
    quizid_left = db.Column(db.Integer)
    subcategory_left = db.Column(db.String(200))
    ctr_left = db.Column(db.Integer)
```

All the attributes ending with _left help in state restoring which we shall explain next.

3) The Quizzes :’)

So you can choose whatever subcategory you like and start attempting question. There are two kinds: MCQ with one correct and MAMCQ with more than one correct as well. There is no negative marking and one point for each correct answer. Oh also, there is a 50-50 LIFELINE which you can use to narrow your options down to two. But a correct answer would only fetch you 0.5 marks. The lifeline part was a little tricky to implement. We had to go back into the database, fetch the correct answer and any other one, and display them only.

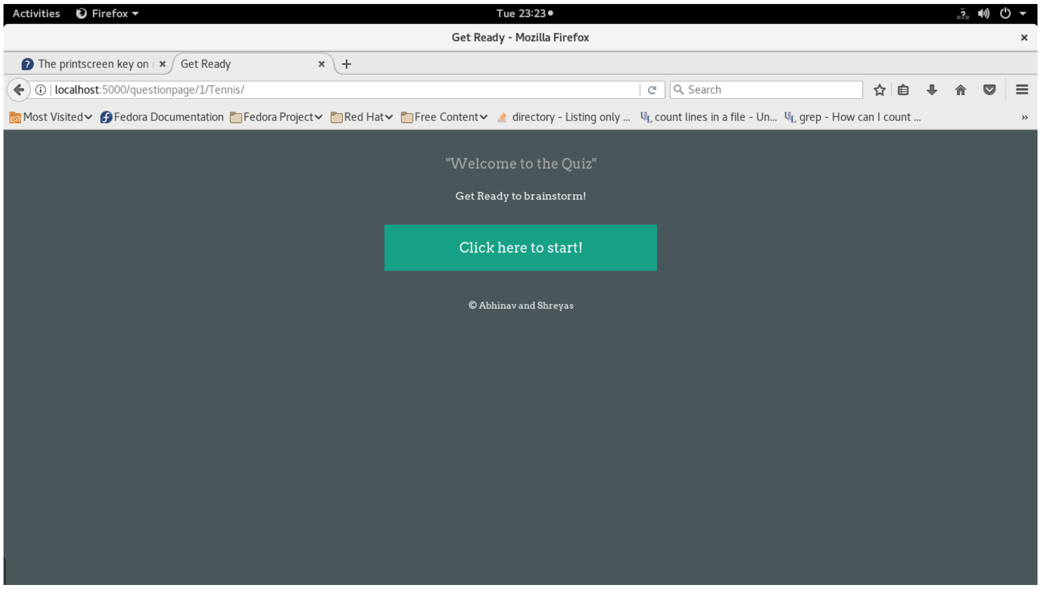
```
obj= Quizzes.query.filter_by(id=ctr).first()
```

The data in the tables was of the type: unicode. So we had to convert it so that we could render it in the html page with our Jinja2 template. We used this command for that:

```
question=unicodedata.normalize('NFKD',obj.question).encode('ascii','ignore')
```

Oh yes we forgot about the Quizzes table! That is the table which contains all of the questions, the categories they belong to, their options and the answer!

```
class Quizzes(db.Model):
        __tablename__ = 'quizzes'

        id=db.Column(db.Integer, primary_key=True, autoincrement=True)

        quiz_id = db.Column(db.Integer)
        category = db.Column(db.String(200))
        sub_id = db.Column(db.Integer)
        subcategory = db.Column(db.String(200))

        question = db.Column(db.String(200))
        opt1=db.Column(db.String(200))
        opt2=db.Column(db.String(200))
        opt3=db.Column(db.String(200))
        opt4=db.Column(db.String(200))

        typeofquestion=db.Column(db.Integer)

        answer=db.Column(db.Integer)
```

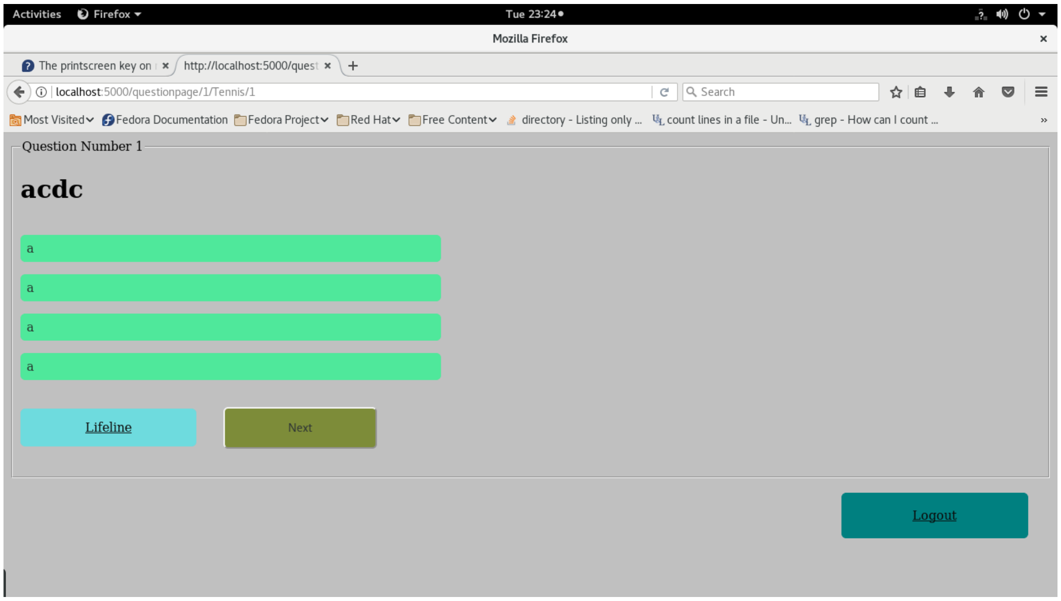


Activities
Firefox
Tue 23:23
Get Ready - Mozilla Firefox
The printscreen key on
Get Ready
localhost:5000/questionpage/1/Tennis/
Search
Most Visited
Fedora Documentation
Fedora Project
Red Hat
Free Content
directory - Listing only ...
count lines in a file - Un...
grep - How can I count ...

"Welcome to the Quiz"

Get Ready to brainstorm!

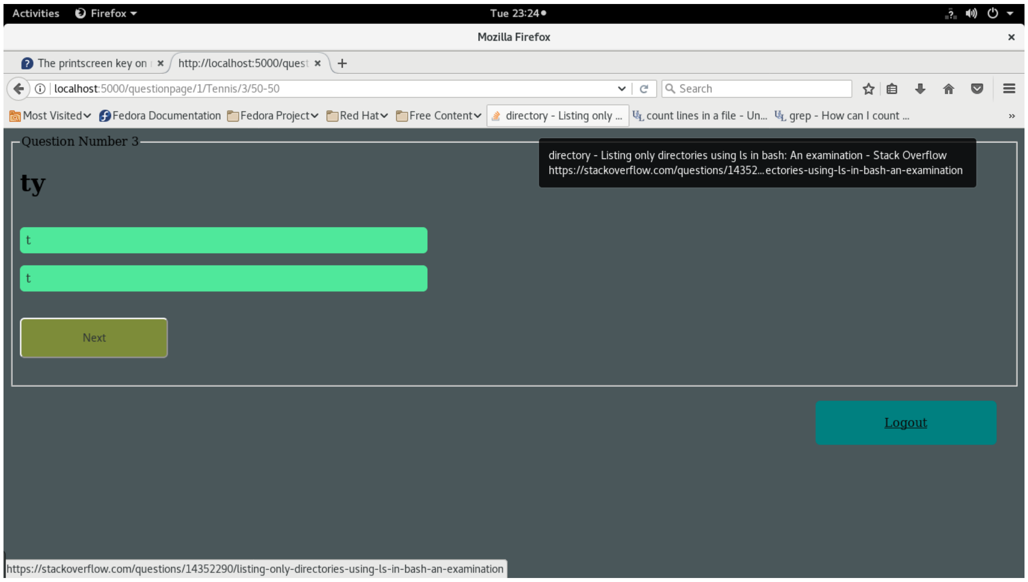
© Abhinav and Shreyas



Activities
Firefox
Tue 23:24
Mozilla Firefox
The printscreen key on
http://localhost:5000/quest
localhost:5000/questionpage/1/Tennis/1
Search
Most Visited
Fedora Documentation
Fedora Project
Red Hat
Free Content
directory - Listing only ...
count lines in a file - Un...
grep - How can I count ...

Question Number 1

acdc



Activities
Firefox
Tue 23:24
Mozilla Firefox
The printscreen key on
http://localhost:5000/quest
localhost:5000/questionpage/1/Tennis/3/50-50
Search
Most Visited
Fedora Documentation
Fedora Project
Red Hat
Free Content
directory - Listing only ...
count lines in a file - Un...
grep - How can I count ...

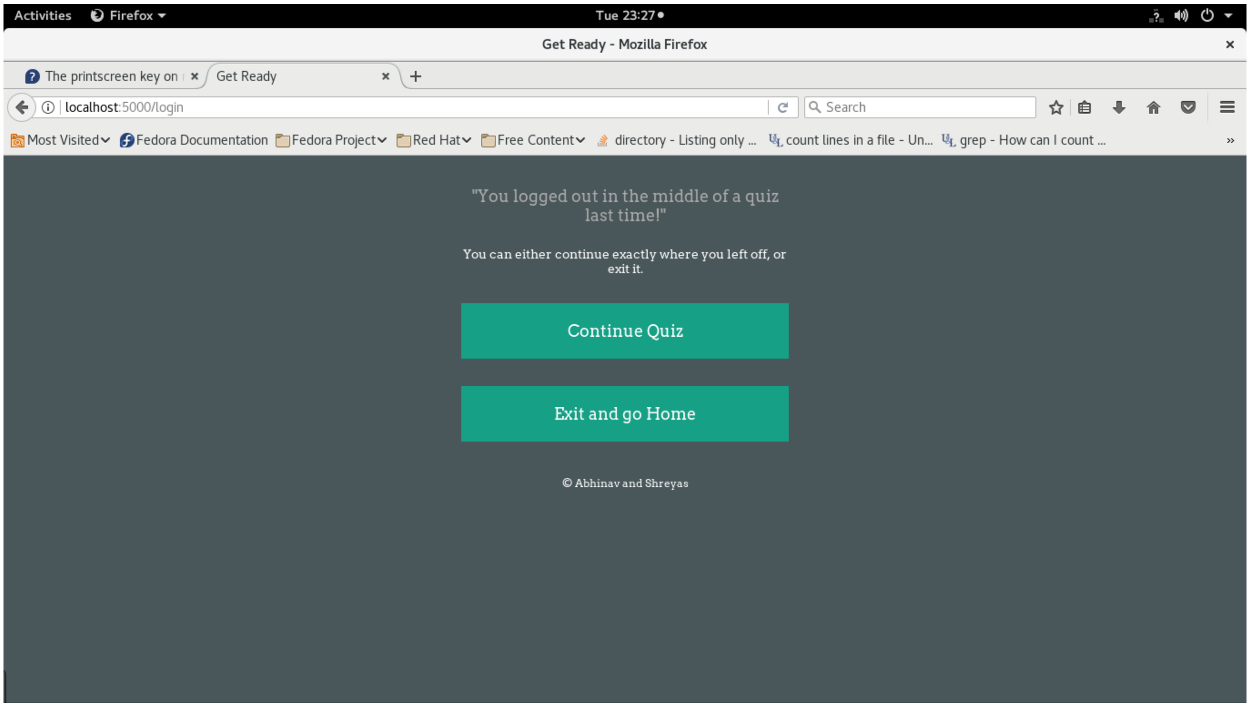
Question Number 3

directory - Listing only directories using ls in bash: An examination - Stack Overflow
https://stackoverflow.com/questions/14352...ectories-using-ls-in-bash-an-examination

ty

https://stackoverflow.com/questions/14352290/listing-only-directories-using-ls-in-bash-an-examination

We also have an option in the middle of a Quiz! So when we log back in, we can either start over with a new quiz or we can CONTINUE FROM WHERE WE LEFT OFF!! This is the state restoring part for which we had the _left attributes in the Scores table.

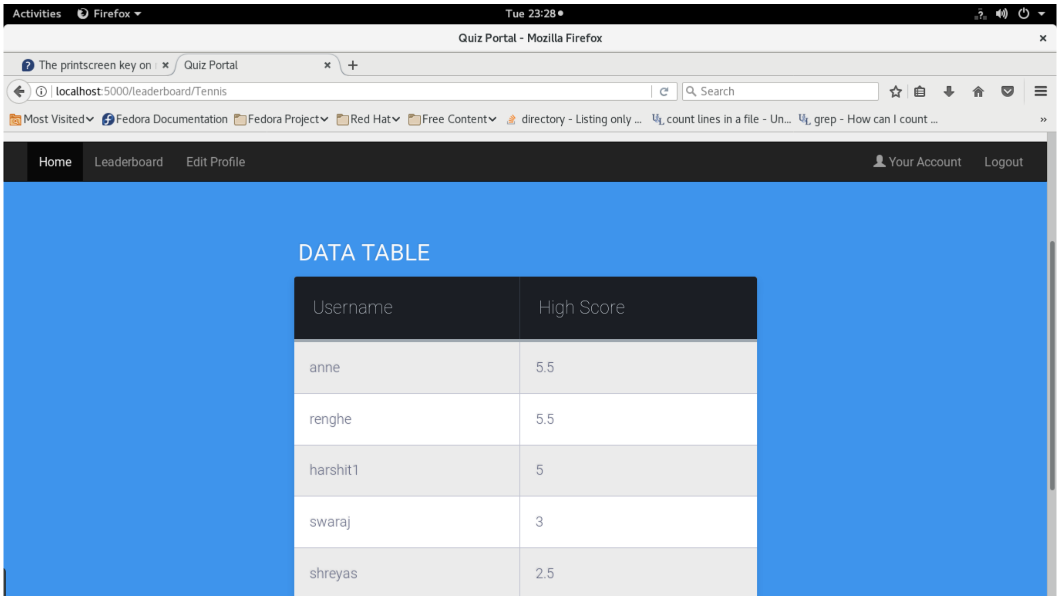


"You logged out in the middle of a quiz last time!"

You can either continue exactly where you left off, or exit it.

© Abhinav and Shreyas

At the end of a quiz, we can check out the leaderboard for that particular quiz. We can see which user has the highest score on that quiz.



DATA TABLE

Username	High Score
anne	5.5
renghe	5.5
harshit1	5
swaraj	3
shreyas	2.5

This was implemented with a easy line of code:

```
haha=Scores.query.filter_by(subcategory=subway).filter(Scores.username !=
"admin").order_by(desc(Scores.high_score))
```

We took all the data of users who attempted that particular quiz and sorted them in descending order!

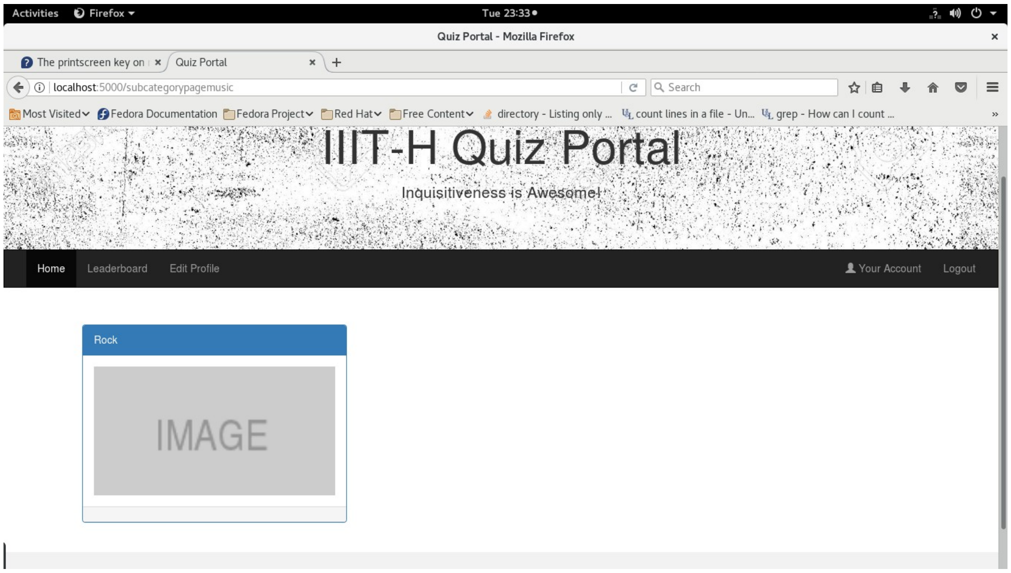
4) We also implemented “Edit Profile” that would let the user change his/her username and password. We took the new ones as inputs into a form. We checked if that username already exists. Then we went into all the databases and updated the username of that person!

```
obj = s.query(User).filter_by(username=USERNAME_NEW).first()
        if obj is None:

                cur_user=s.query(User).filter_by(username=loggedinuser).first()
                cur_user.username = USERNAME_NEW
                cur_user.password = PASS_NEW
                s.commit()

                obj2=Scores.query.filter_by(username=loggedinuser).all()
                for i in obj2:
                        i.username=USERNAME_NEW
                db.session.commit()
```

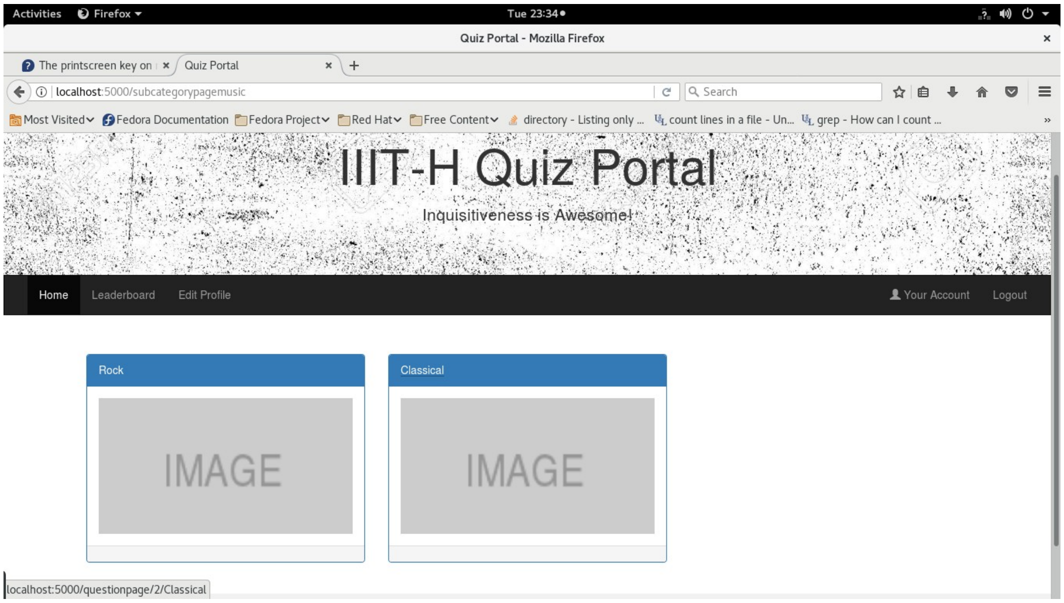
5) The Admin Account:

We are the bosses and have access to the entire database! We can add questions to already existing quizzes or even create a whole new subcategory! That was also a little tricky to implement. But we did not hard code the subcategory quizzes. So we can have as many as we wish.

Before:

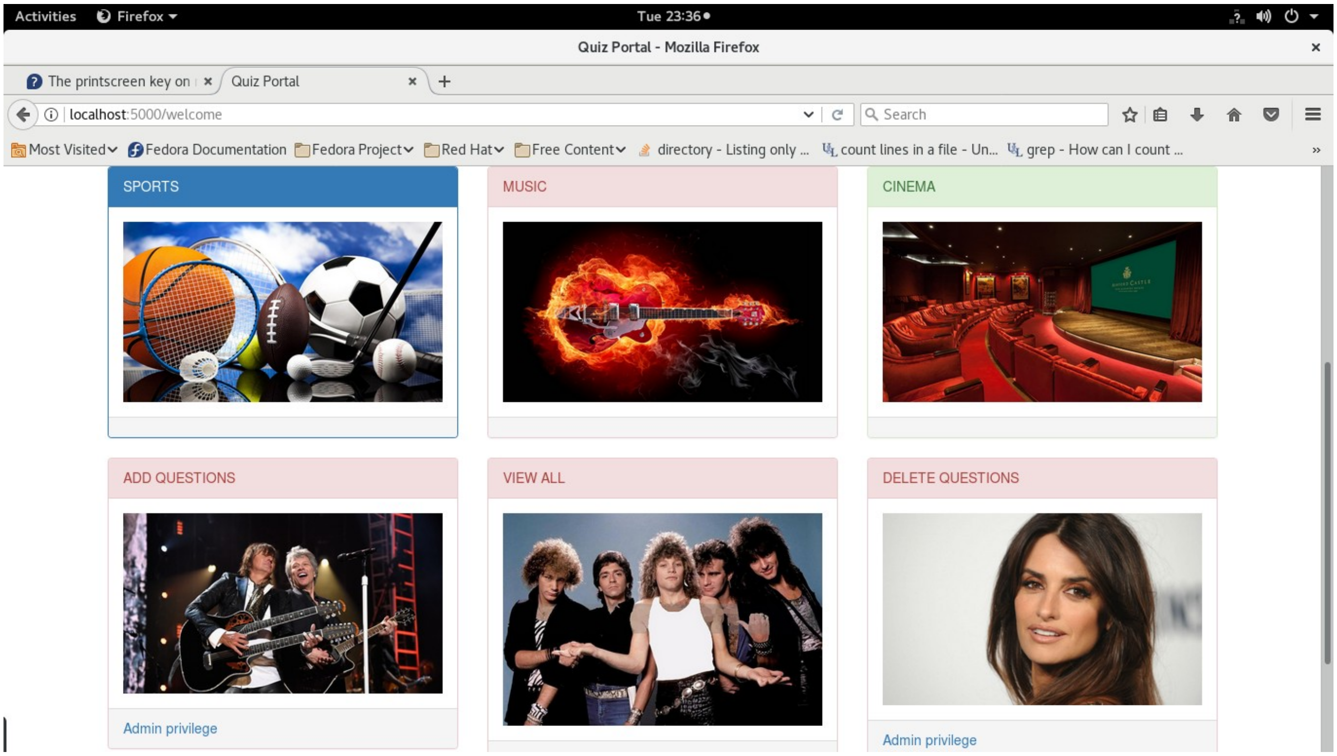


After adding a new question of new subcategory:

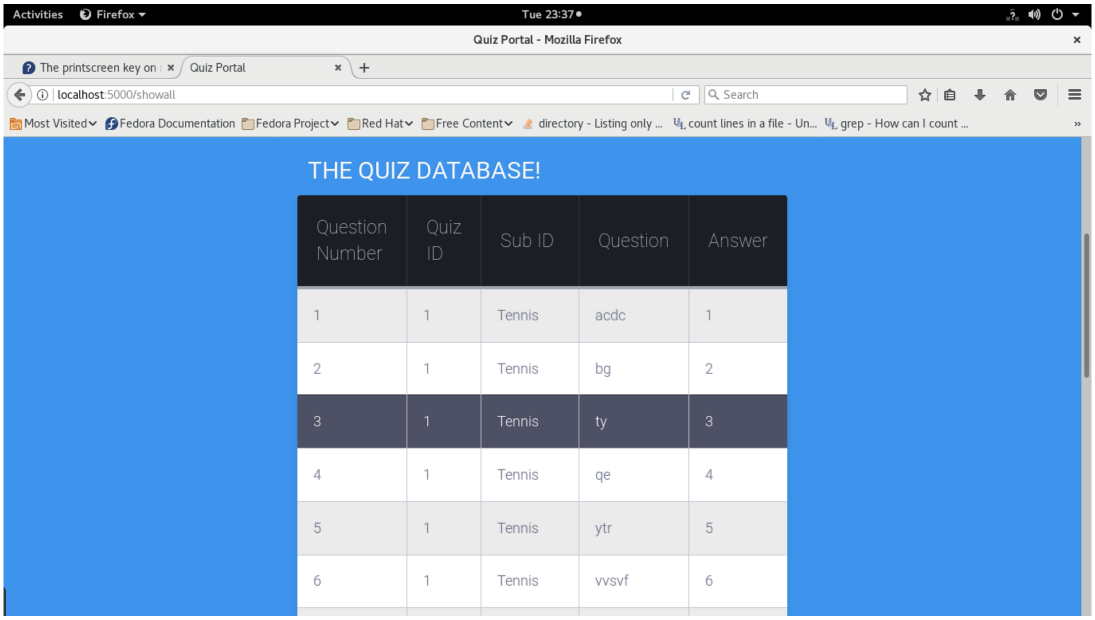


So yes the admin has all these privileges and has access to the entire database and all the tables in it!

Here is how the Admin home page looks like:



And the entire table with all questions and answers!



THE QUIZ DATABASE!

Question Number	Quiz ID	Sub ID	Question	Answer
1	1	Tennis	acdc	1
2	1	Tennis	bg	2
3	1	Tennis	ty	3
4	1	Tennis	qe	4
5	1	Tennis	ytr	5
6	1	Tennis	vvsvf	6

So yes that’s about it! Hope you’ve understood this. We have the code put up on our github, so you can take a look and we’d be glad if you could send us pull requests if you figure out some bug. If you have any questions whatsoever, please feel free to send a mail to us.

Abhinav: abhinav.g@students.iiit.ac.in
Shreyas: shreyas.shankar@students.iiit.ac.in

That is our IIIT-H Quiz Portal, the first web app created by us. Have fun Quizzing; hope you enjoy using it as much as we did building it :D

-Abhinav and Shreyas