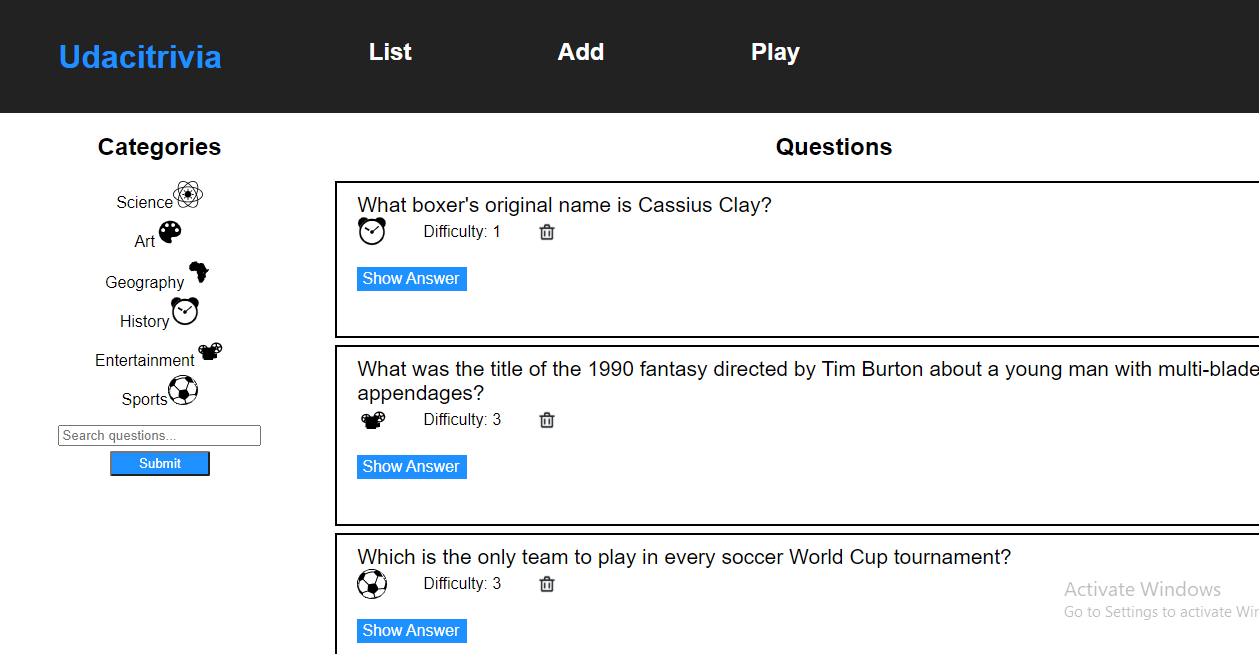
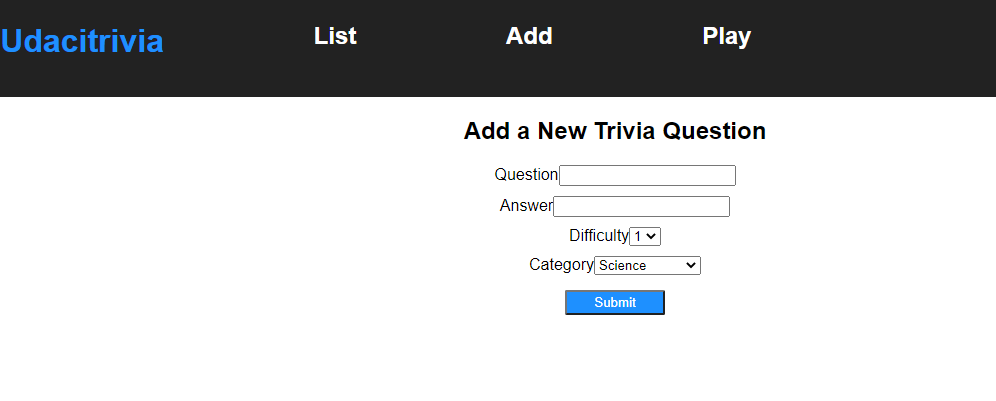
* **Project Title:**

**Trivia ptoject is a collection of questions added by other and any one can play game to answer on any question. He select the category that he want to answer its random question and get his quize score .**

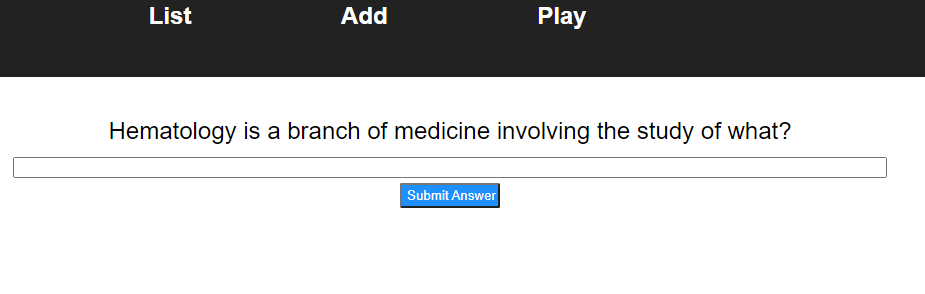
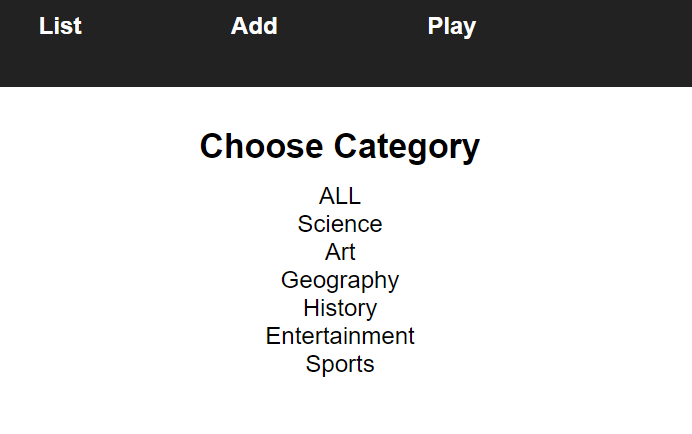
**Questions page:**



**Add question:**



**Play:**



* **Pre-requisites and Local Development:**

Developers using this project should already have Python3, pip and node installed on their local machines.

#### Backend:

From the backend folder run *pip install requirements.txt*. All required packages are included in the requirements file.

To run the application run the following commands:

*export FLASK\_APP=flaskr*

*export FLASK\_ENV=development*

*flask run*

These commands put the application in development and directs our application to use the *\_\_init\_\_.py* file in our flaskr folder. Working in development mode shows an interactive debugger in the console and restarts the server whenever changes are made. If running locally on Windows, look for the commands in the [Flask documentation](http://flask.pocoo.org/docs/1.0/tutorial/factory/).

The application is run on *http://127.0.0.1:5000/* by default and is a proxy in the frontend configuration.

* **Frontend:**

From the frontend folder, run the following commands to start the client:

*npm install // only once to install dependencies*

*npm start*

By default, the frontend will run on *localhost:3000.*

### **Tests:**

In order to run tests navigate to the backend folder and run the following commands:

*dropdb trivia \_test*

*createdb trivia \_test*

*psql trivia\_test < trivia.psql*

*python test\_flaskr.py*

The first time you run the tests, omit the dropdb command.

All tests are kept in that file and should be maintained as updates are made to app functionality.

### **Error Handling:**

Errors are returned as JSON objects in the following format:

{

"success": False,

"error": 400,

"message": "bad request"

}

The API will return three error types when requests fail:

* 400: Bad Request
* 404: Resource Not Found
* 422: Not Processable

## **API Reference:**

<https://flask-restful.readthedocs.io/en/latest/>

## **Authors:**

Udacity academy.

## **Acknowledgements:**

Leader team in fwd and other studens .