PROJECT REPORT ON SUMMER INDUSTRIAL TRAINING

COURSE TITLE :

WEB DEVELOPMENT WITH PYTHON & DJANGO

(AUGUST 14th, 2019 to SEPTEMBER 14th ,2019)

PROJECT TITLE :

QBRAIN

Submitted By:

Debanjan Bhattacharyya *(Roll No. : 16800116076)*

(CSE, 4th Year, Saroj Mohan Institute Of Technology)

TABLE OF CONTENTS

ACKNOWLEDGEMENT

This Project Titled Question Submitting Website (QBrain) was supported by WEBTEK LABS PVT. LTD. .

We are thankful to our mentor Ms. Deepa Ashok Kumar ,who provided expertise that greatly assisted the project completion. We are thankful to WEBTEK LABS for giving us an opportunity to work there and help us to gain real time job experience.

DECLARATION

We, Debanjan Bhattacharyya, Avik Ghosh, Rajat Saha, students of *CSE*, *4th Year* of *<Saroj Mohan Institute Of Technology>* *<Kolkata>* *<West Bengal>* have completed summer training project from Webtek Labs Pvt. Ltd. (Duration: 14th August 2019 – 14th September 2019).

During the mentioned period we worked on Web Development with Django and completed the summer training under the guidance of Ms. Deepa Ashok Kumar.

Debanjan Bhattacharyya:

Avik Ghosh:

Rajat Saha:

DATE : _____

INTRODUCTION

Web development is a broad term for the work involved in developing a web site for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing the simplest static single page of plain text to the most complex web- based internet applications, electronic businesses, and social network services. A more comprehensive list of tasks to which web development commonly refers, may include web engineering, web design, web content development, client liaison, client-side/server-side scripting, web server and network security configuration, and e-commerce development. For larger organizations and businesses, web development teams can consist of hundreds of people(web developers) and follow standard methods like Agile methodologies while developing websites. Smaller organizations may only require a single permanent or contracting developer, or secondary assignment to related job positions such as a graphic designer and/or information systems technician. Web development may be a collaborative effort between departments rather than the domain of a designated department. There is open source software for web development like Django(web_framework),GlassFish, LAMP (Linux, Apache, MySQL, PHP) stac k and Perl/Plack. This has kept the cost of learning web development to a minimum. Another contributing factor to the growth of the industry has been the rise of easy-to-use WYSIWYG web-development software, such as Adobe Dreamweaver , Blue Griffon and Microsoft Visual Studio. Knowledge of Hyper Text Markup Language (HTML) or of programming languages is still required to use such software, but the basics can be learned and implemented quickly with the help of help files, technical books, internet tutorials, or face-to-face training.

TECHNOLOGIES USED

❖ Text editor software and local server installation :

- ✓ Details about the software requirement
- ✓ Download and installation process
- ✓ Download and installation of Python, Django, Notepad++
- ✓ Setting up database and creating database connections

❖ HTML Layouts :

- ✓ Div containers
- ✓ Form fields
- ✓ Tables
- ✓ Buttons

❖ CSS Designing :

- ✓ Navigation bar
- ✓ Footer
- ✓ Background images
- ✓ Div containers
- ✓ Responsiveness of overall website
- ✓ Buttons, Fields

❖ JAVASCRIPT Used :

- ✓ Responsiveness of overall website

❖ DJANGO Used :

- ✓ Connecting database
- ✓ Passing and retrieving values from database
- ✓ Server and Client side validations
- ✓ Displaying profile details
- ✓ Search
- ✓ Sessions
- ✓ Validations on signup and login pages

❖ Other Components Used :

- ✓ Bootstrap

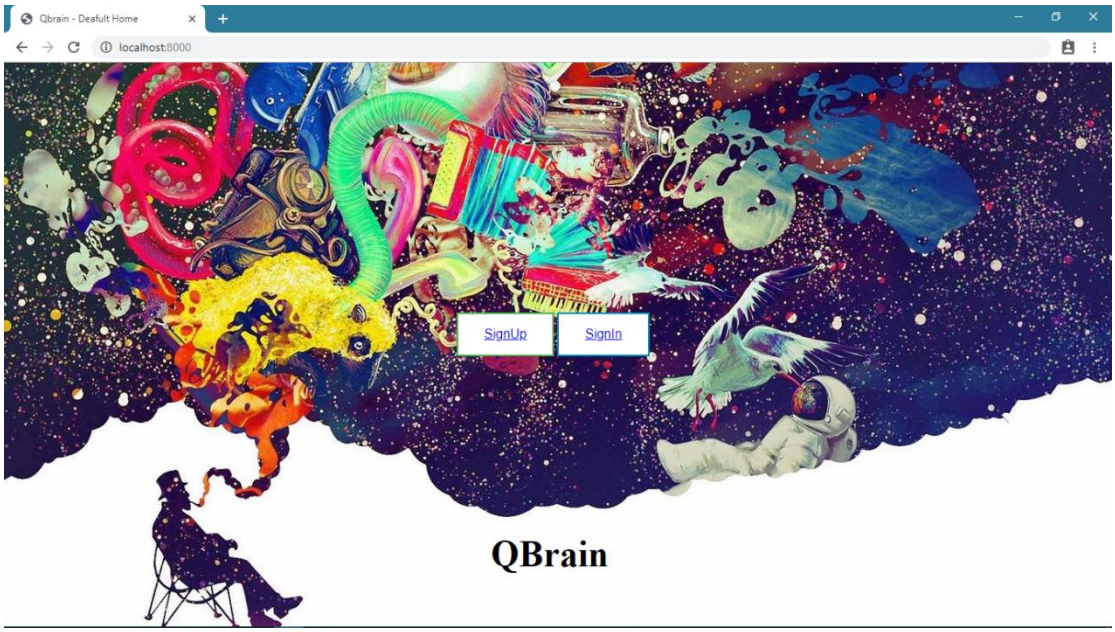
DESCRIPTION OF THE PROJECT

Humans are curious by nature and we have thousands of queries in our brain. QBrain is an online app where you can ask your queries and get answered by other users. Everything is saved in our database so don’t worry about your question getting deleted, and you may also answer other’s questions. All the questions are categorised by there categories and while answering others you can go to your category and answer.

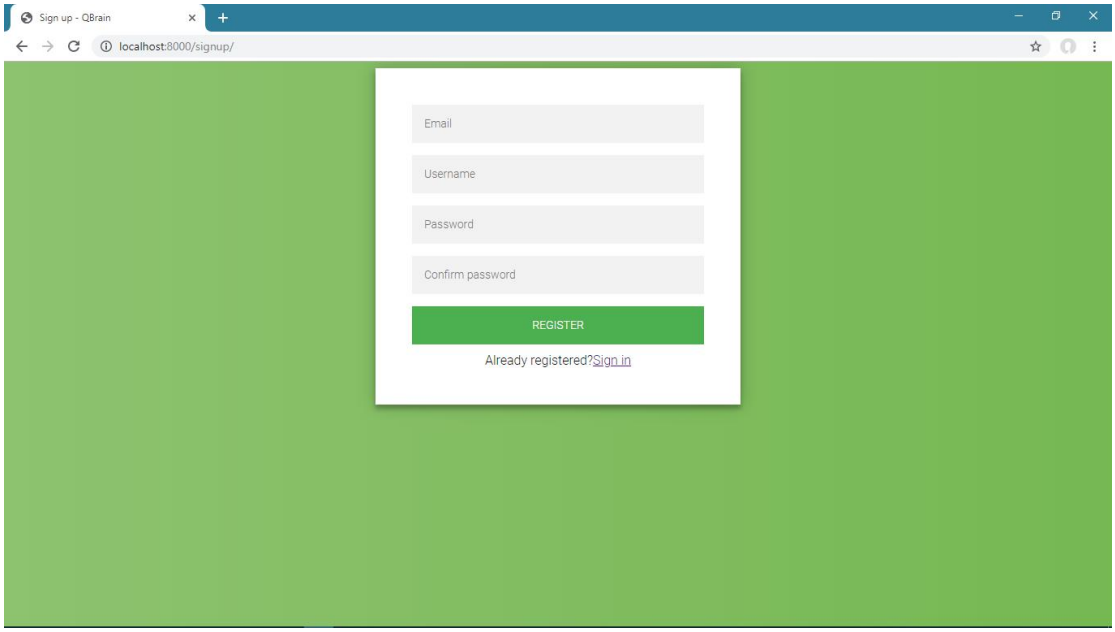
I made this project during the training I did few months back on web development using Django at WebTek.

SnapShots

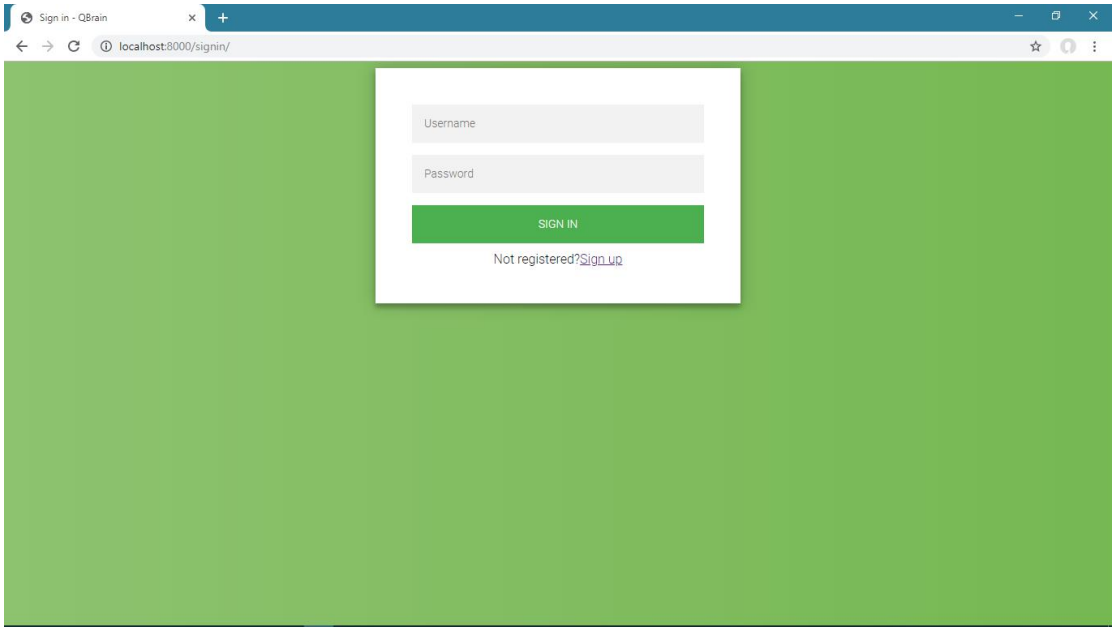
LANDING PAGE: CLICK ON SIGN UP IF YOU'RE NEW USER ELSE ON SIGN IN.



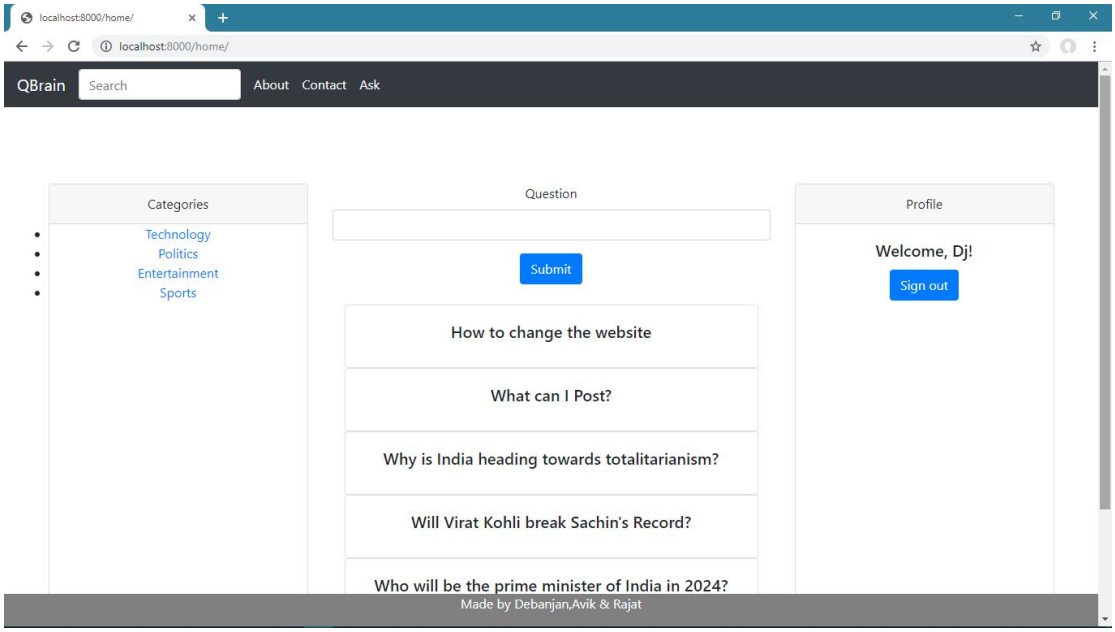
SIGN UP PAGE: REGISTER YOURSELF BY GIVING THE NEEDED INFORMATION



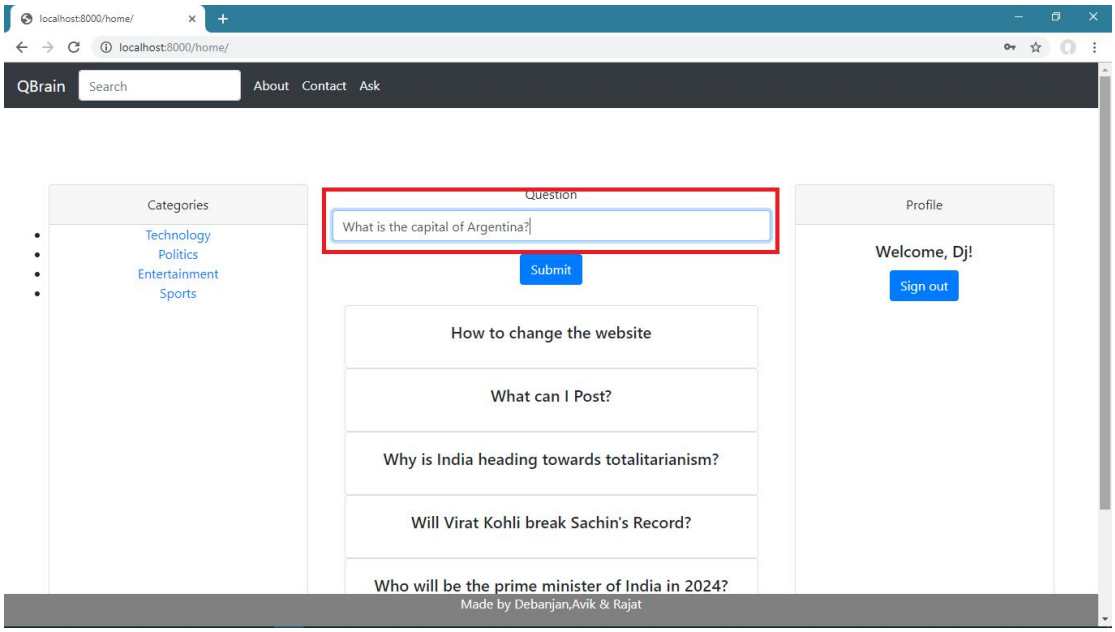
SIGN IN PAGE: CLICK ON SIGN IN IF YOU'RE ALREADY AN USER.



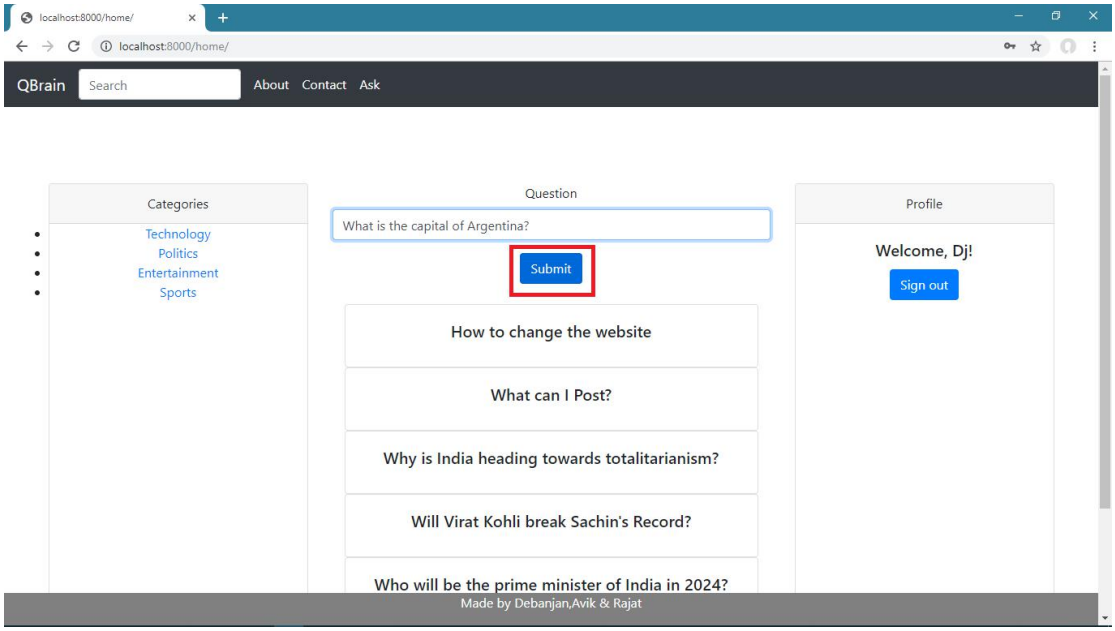
HOME PAGE: AFTER SUCCESFULLY LOGGING IN YOU WILL COME TO THI PAGE



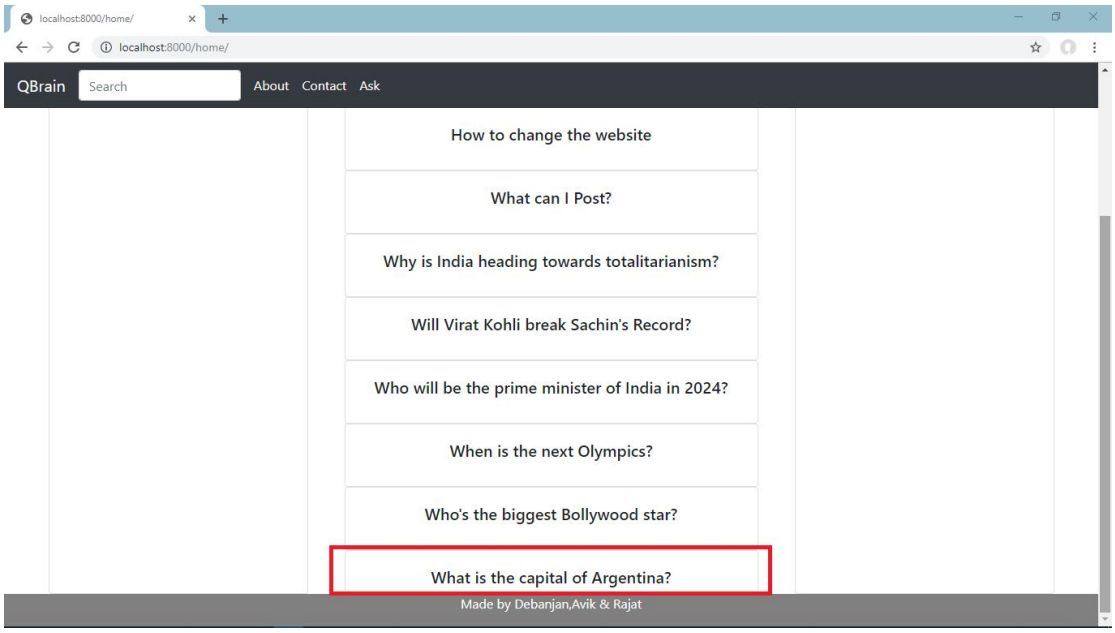
IN THE QUESTION BOX YOU CAN ASK YOUR QUESTION



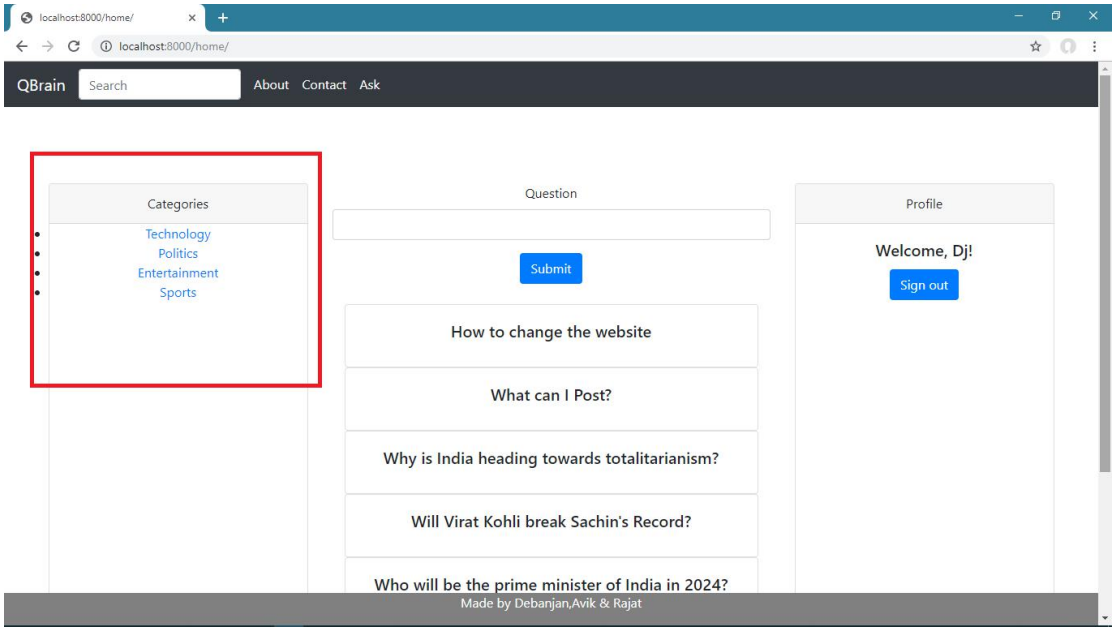
CLICK ON THE SUBMIT BUTTON TO POST YOUR QUESTION



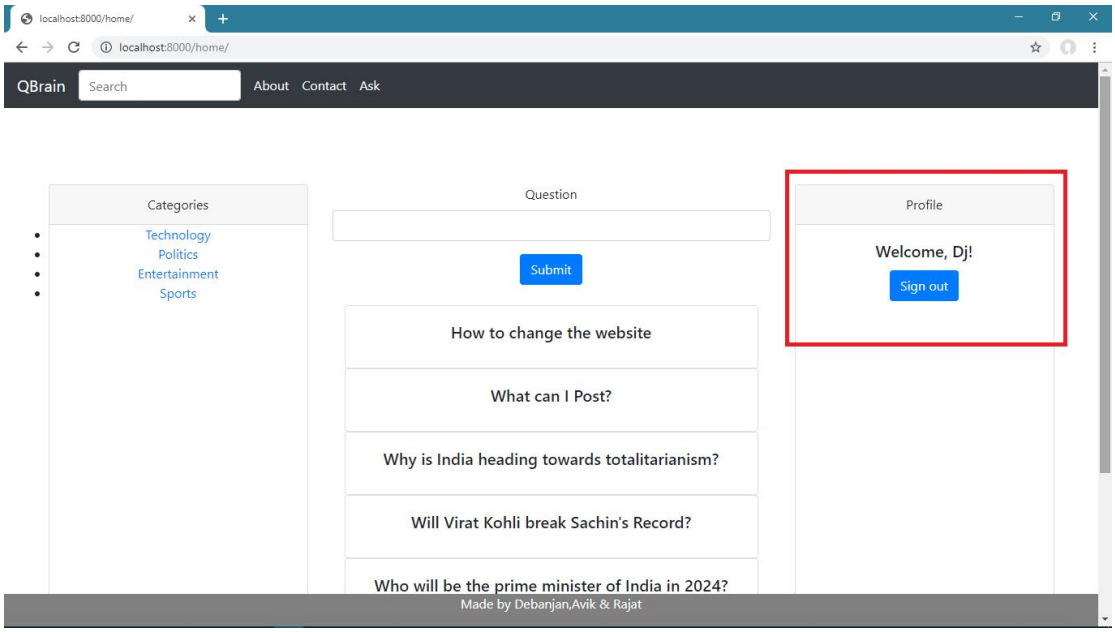
YOU CAN SEE YOUR QUESTION AT THE BUTTOM



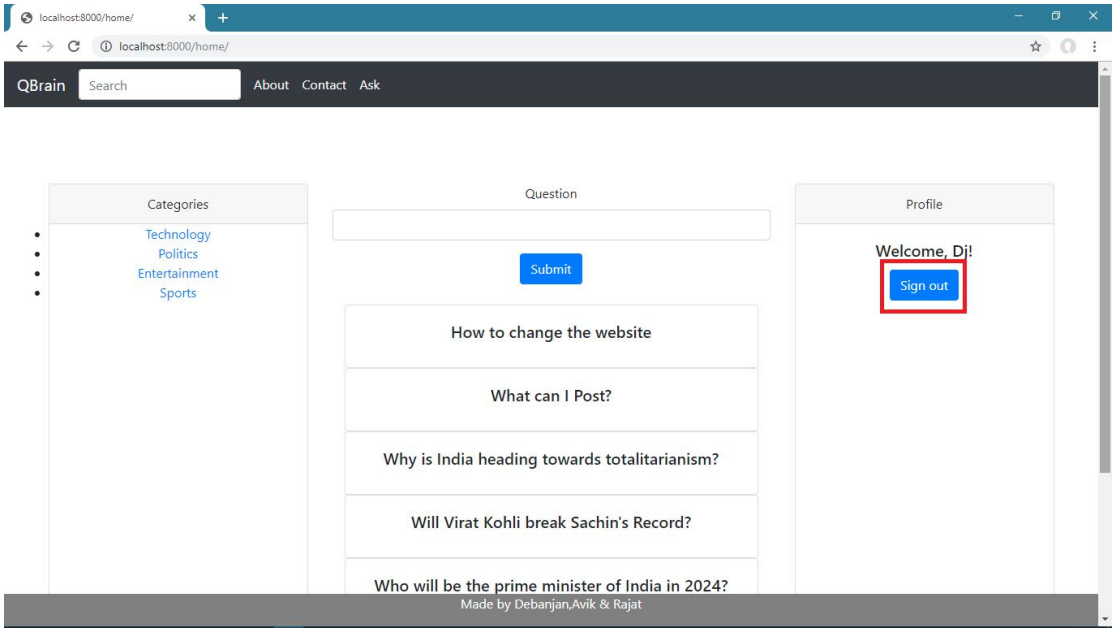
HERE ARE THE CATEGORIES



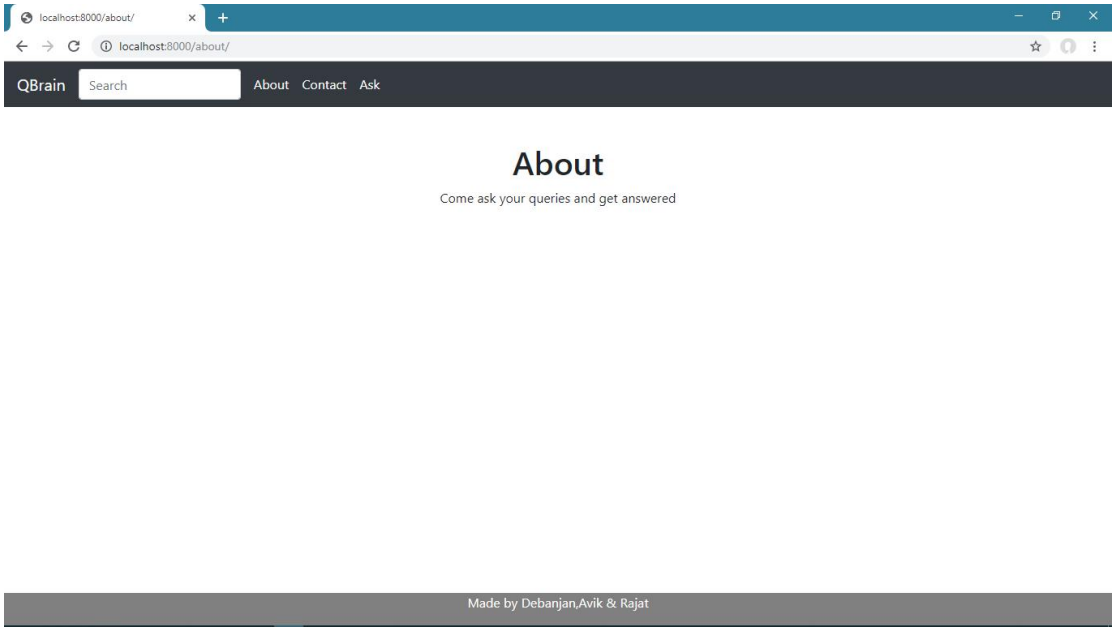
YOUR USERNAME WITH AN WELCOME MESSAGE WILL BE HERE



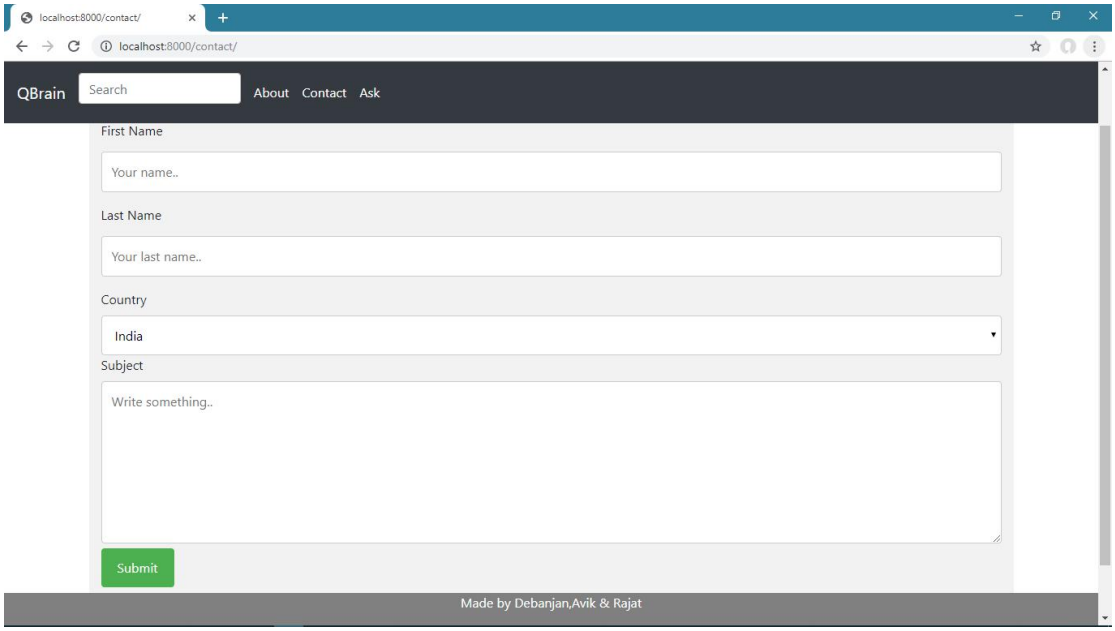
BY CLICKING ON SIGN OUT YOU WILL BE LOGGED OUT OF YOUR ACCOUNT



ABOUT PAGE: *Come ask your queries and get answered*



CONTACT PAGE: YOU CAN SEND US YOUR FEEDBACK OR ANY QUESTION ABOUT OUR WEBSITE



VIEWS.PY

views.py - D:\QBrain\qbrainp\qbrainapp\views.py (3.7.3)

File Edit Format Run Options Window Help

```
from django.shortcuts import render,redirect
from qbrainapp.models import category,qbraindb
from qbrainapp.forms import qbraindbForm,UserForm
from django.contrib.auth.models import User
from django.contrib.auth import authenticate,login,logout
#from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.decorators import login_required


def landing(request):
    return render(request, "qbrain/landing-page.html")

""" def home(request):
    categories=category.objects.all()
    return render(request,"qbrain/home.html",{'categories':categories}) """

@login_required
def home(request):
    if request.method == "POST":
            form = qbraindbForm(request.POST)
            print("hhhhhhhhhhhh")
            if form.is_valid():
                print("yyyyyyyyyyy")
                new = form.save(commit=False)
                new.author = request.user
                new.save()
                #form.save()
                return redirect("home")
    else:
        form = qbraindbForm()
        categories = category.objects.all()
```

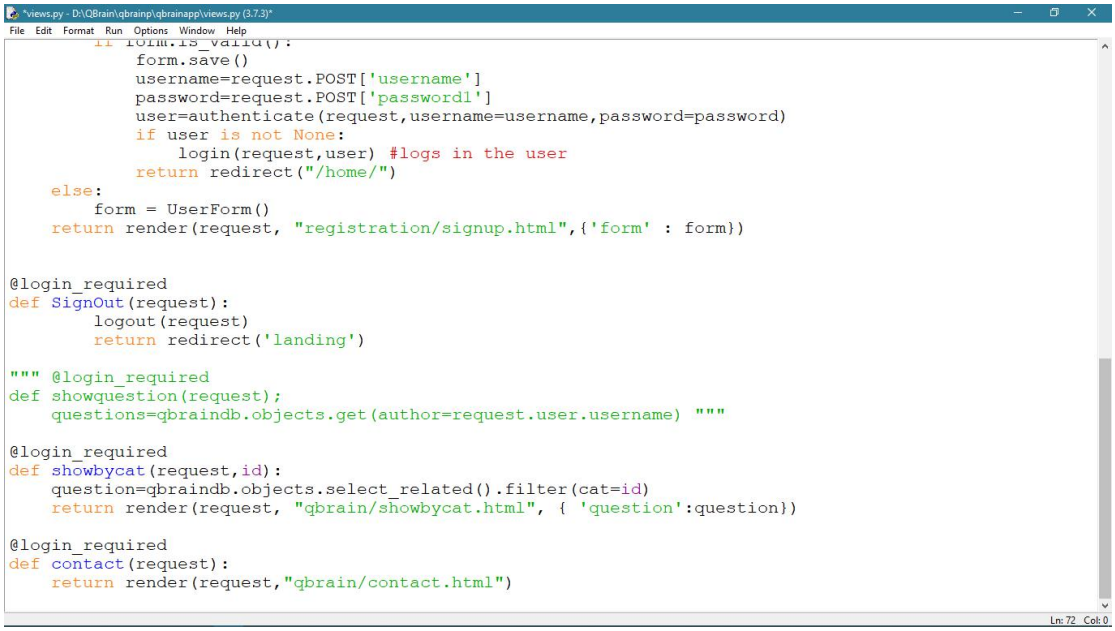
Ln: 8 Col: 0

views.py - D:\QBrain\qbrainp\qbrainapp\views.py (3.7.3)

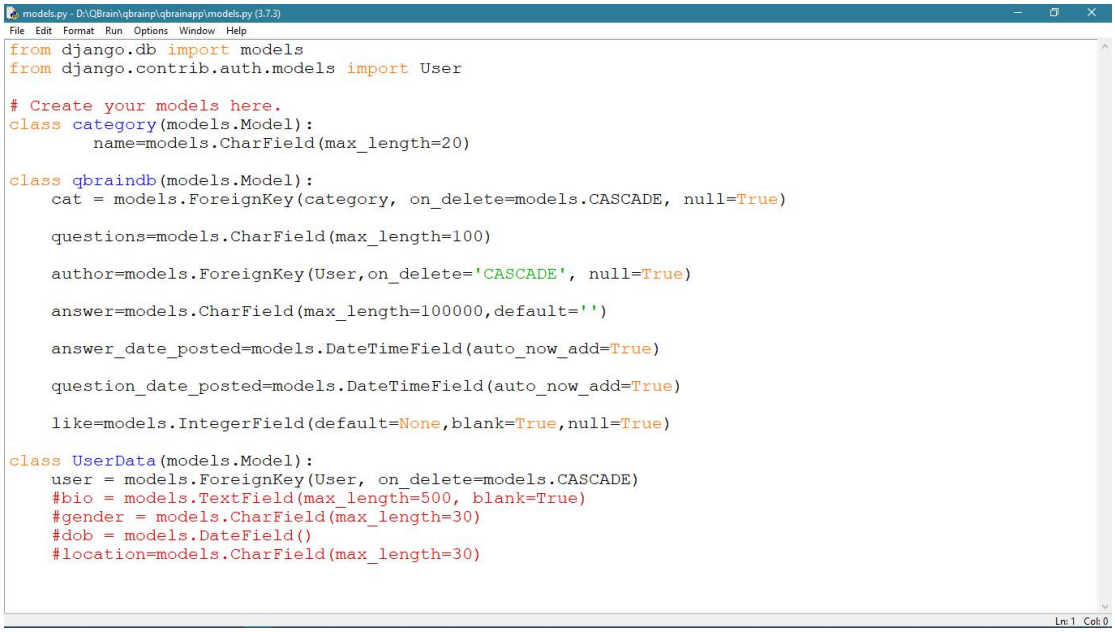
File Edit Format Run Options Window Help

```
                return redirect("home")
    else:
        form = qbraindbForm()
        categories = category.objects.all()
        questions = qbraindb.objects.all()
        return render(request, "qbrain/home.html", {'form': form, 'categories': categories, 'questions'


def about(request):
    return render(request,"qbrain/about.html")

def signup(request):
    if request.method=="POST":
        form = UserForm(request.POST)
        if form.is_valid():
            form.save()
            username=request.POST['username']
            password=request.POST['password1']
            user=authenticate(request,username=username,password=password)
            if user is not None:
                login(request,user) #logs in the user
            return redirect("/home/")
    else:
        form = UserForm()
    return render(request, "registration/signup.html",{'form' : form})


@login_required
def SignOut(request):
        logout(request)
        return redirect('landing')
```

Ln: 57 Col: 0


views.py - D:\QBrain\qbrainp\qbrainapp\views.py (3.7.3)
File Edit Format Run Options Window Help
if form.is_valid():
form.save()
username=request.POST['username']
password=request.POST['password1']
user=authenticate(request,username=username,password=password)
if user is not None:
login(request,user) #logs in the user
return redirect("/home/")
else:
form = UserForm()
return render(request, "registration/signup.html",{'form' : form})
@login_required
def SignOut(request):
logout(request)
return redirect('landing')
""" @login_required
def showquestion(request);
questions=qbraindb.objects.get(author=request.user.username) """
@login_required
def showbycat(request,id):
question=qbraindb.objects.select_related().filter(cat=id)
return render(request, "qbrain/showbycat.html", { 'question':question})
@login_required
def contact(request):
return render(request,"qbrain/contact.html")
Ln: 72 Col: 0

MODELS.PY


models.py - D:\QBrain\qbrainp\qbrainapp\models.py (3.7.3)
File Edit Format Run Options Window Help
from django.db import models
from django.contrib.auth.models import User
# Create your models here.
class category(models.Model):
name=models.CharField(max_length=20)
class qbraindb(models.Model):
cat = models.ForeignKey(category, on_delete=models.CASCADE, null=True)
questions=models.CharField(max_length=100)
author=models.ForeignKey(User,on_delete='CASCADE', null=True)
answer=models.CharField(max_length=100000,default='')
answer_date_posted=models.DateTimeField(auto_now_add=True)
question_date_posted=models.DateTimeField(auto_now_add=True)
like=models.IntegerField(default=None,blank=True,null=True)
class UserData(models.Model):
user = models.ForeignKey(User, on_delete=models.CASCADE)
#bio = models.TextField(max_length=500, blank=True)
#gender = models.CharField(max_length=30)
#dob = models.DateField()
#location=models.CharField(max_length=30)
Ln: 1 Col: 0

FORMS.PY

forms.py - D:\QBrain\qbrainp\qbrainapp\forms.py (3.7.3)

File Edit Format Run Options Window Help

```
from django.forms import ModelForm
from django import forms
from qbrainapp.models import qbraindb,category
from django.contrib.auth.models import User
from qbrainapp.models import UserData
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm


class qbraindbForm(ModelForm):
    class Meta:
        model = qbraindb
        fields = ('questions', )

class UserForm(UserCreationForm):
    class Meta:
        model = User
        fields = ('email','username','password1','password2')
```

Ln: 1 Col: 0

URLS.PY

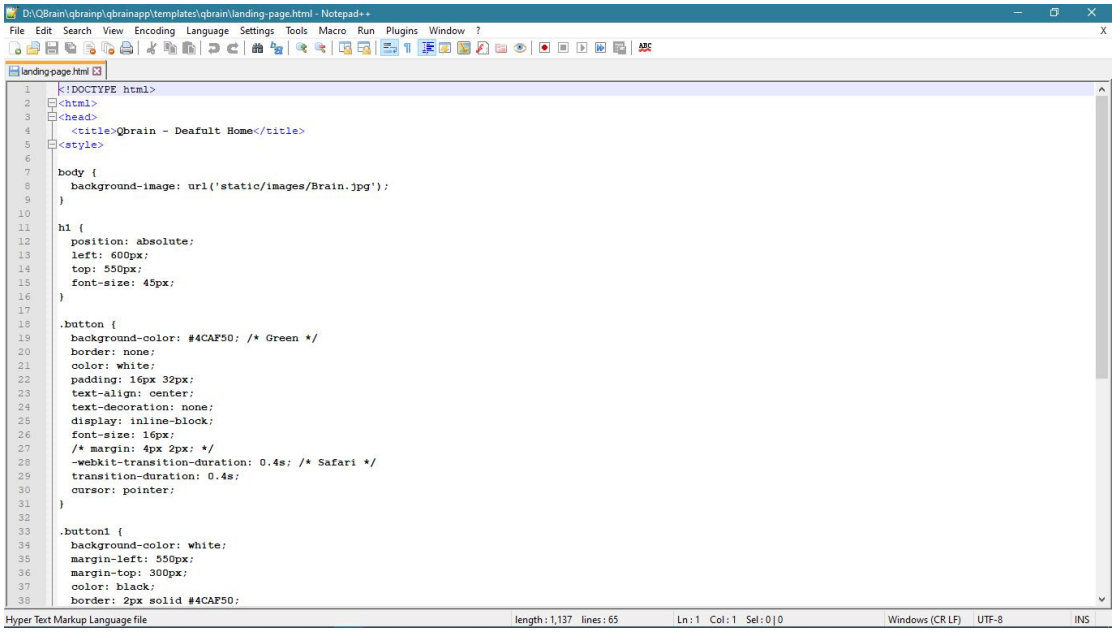
urls.py - D:\QBrain\qbrainp\qbrainp\urls.py (3.7.3)

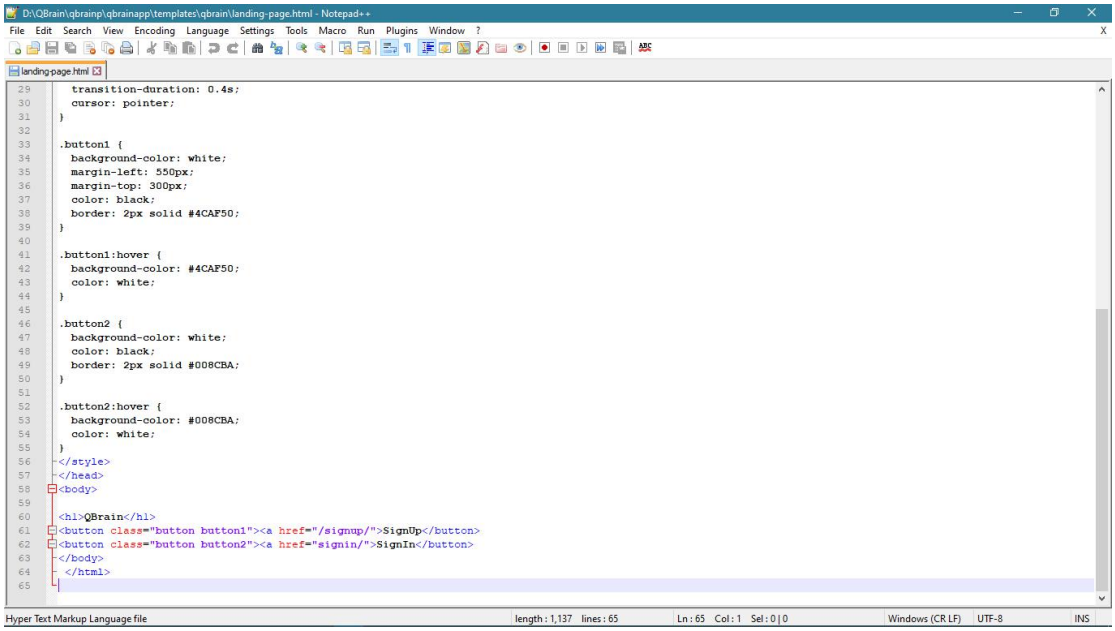
File Edit Format Run Options Window Help

```
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from qbrainapp import views
from django.contrib.auth.views import LoginView
from django.contrib.auth import views as auth_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views.landing,name="landing"),
    path('home/',views.home, name="home"),
    path('showbycat/<int:id>',views.showbycat),
    path('signup/',views.signup,name="signup"),
    path('about/',views.about,name="about"),
    #path('signin/',LoginView.as_view(template_name='diary_app/signin.html', redirect_authenticated_use
    path('signin/', LoginView.as_view(template_name='registration/signin.html', redirect_authenticated_
    path('contact/',views.contact, name="contact"),
    path('logout/',views.SignOut,name="signout")
    # path('signup/',views.signup,name="signup")
]
```

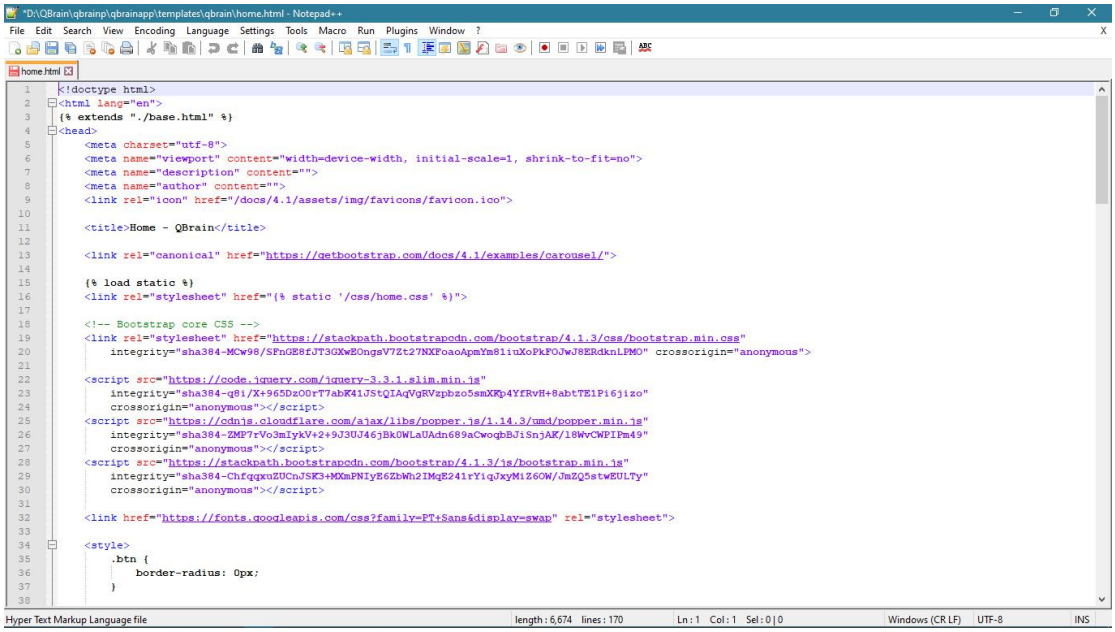
Ln: 23 Col: 0

LANDING-PAGE.HTML



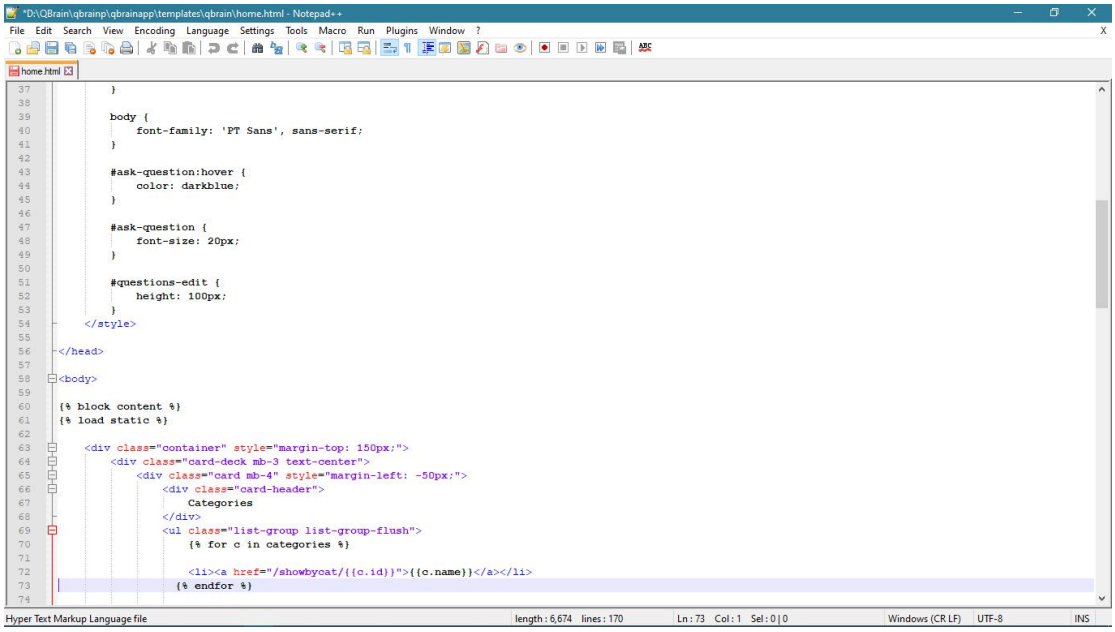


HOME.HTML



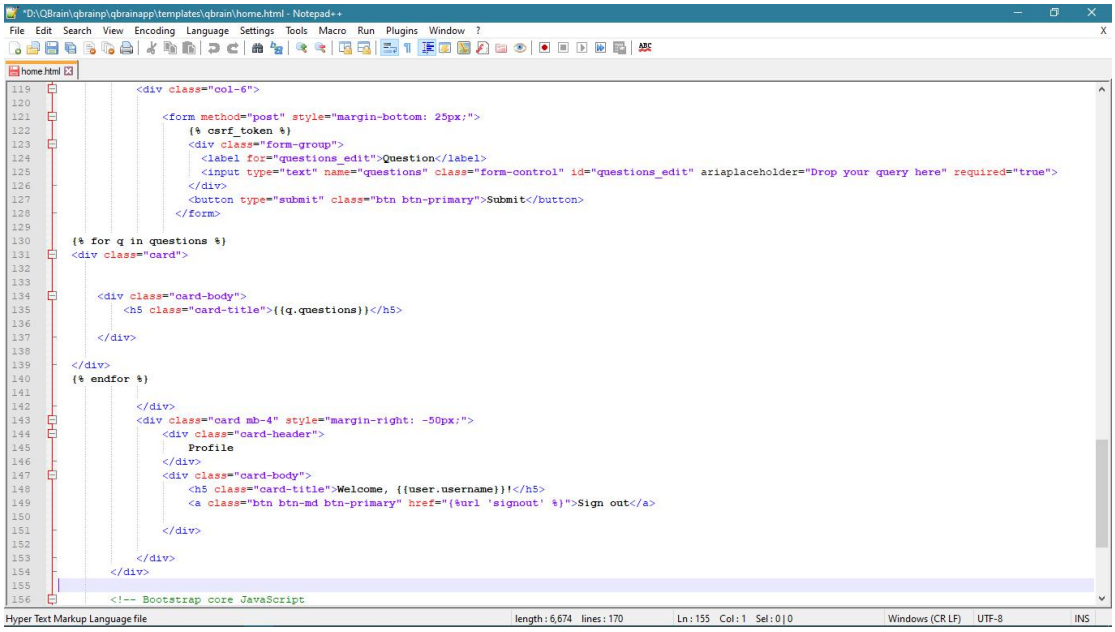
```
<!doctype html>
<html lang="en">
{% extends "./base.html" %}
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="/docs/4.1/assets/img/favicons/favicon.ico">

    <title>Home - QBrain</title>

    <link rel="canonical" href="https://getbootstrap.com/docs/4.1/examples/carousel/">

    {% load static %}
    <link rel="stylesheet" href="{% static '/css/home.css' %}">

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
        integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
        integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
        crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
        integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
        integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
        crossorigin="anonymous"></script>

    <link href="https://fonts.googleapis.com/css?family=PT+Sans&display=swap" rel="stylesheet">

    <style>
        .btn {
            border-radius: 0px;
        }
```

Hyper Text Markup Language file | length : 6,674 lines : 170 | Ln : 1 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

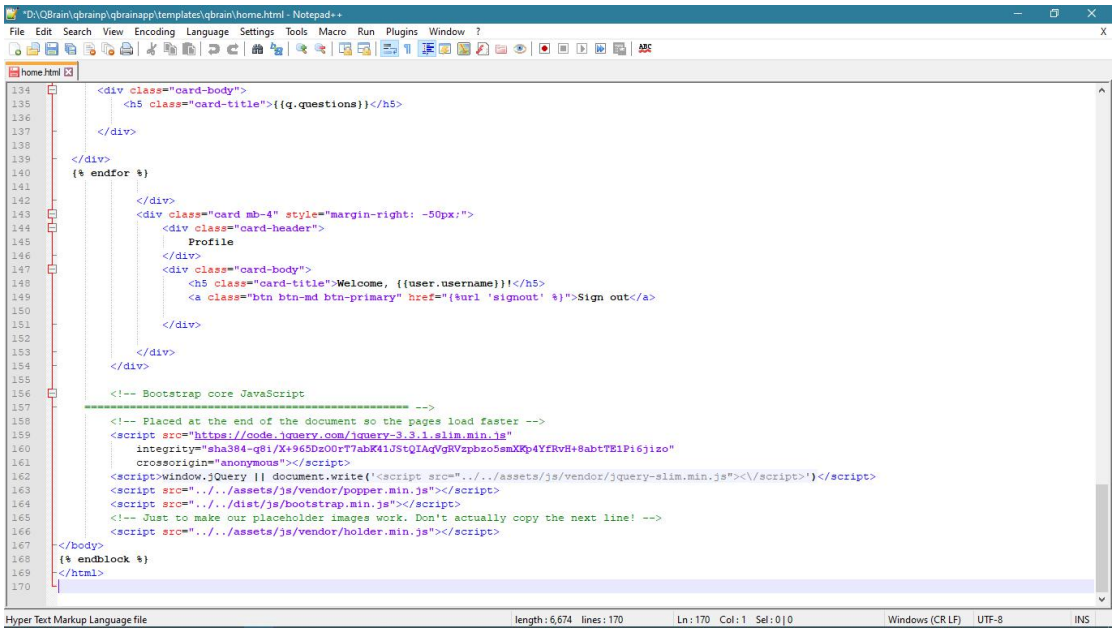


```
        }

        body {
            font-family: 'PT Sans', sans-serif;
        }

        #ask-question:hover {
            color: darkblue;
        }

        #ask-question {
            font-size: 20px;
        }

        #questions-edit {
            height: 100px;
        }
    </style>

</head>

<body>

{% block content %}
{% load static %}

    <div class="container" style="margin-top: 150px;">
        <div class="card-deck mb-3 text-center">
            <div class="card mb-4" style="margin-left: -50px;">
                <div class="card-header">
                    Categories
                </div>
                <ul class="list-group list-group-flush">
                    {% for c in categories %}

                    <li><a href="/showbycat/{{c.id}}">{{c.name}}</a></li>
                    {% endfor %}
```

Hyper Text Markup Language file | length : 6,674 lines : 170 | Ln : 73 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

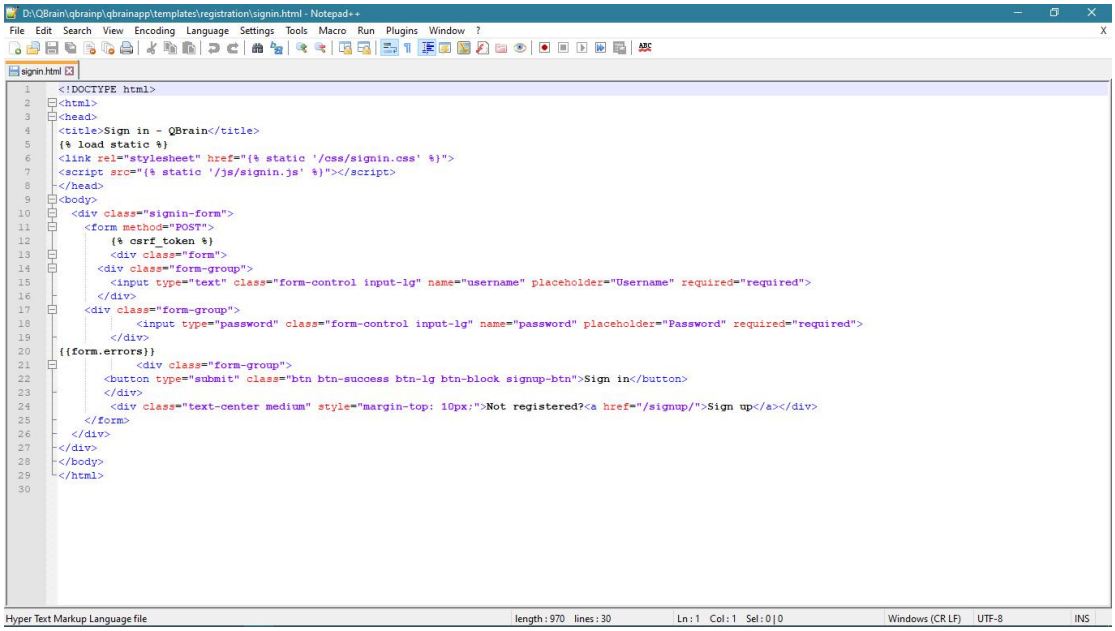


```
*D:\QBrain\qbrainp\qbrainapp\templates\qbrain\home.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
home.html
            <div class="col-6">

                <form method="post" style="margin-bottom: 25px;">
                    {% csrf_token %}
                    <div class="form-group">
                      <label for="questions_edit">Question</label>
                      <input type="text" name="questions" class="form-control" id="questions_edit" ariaplaceholder="Drop your query here" required="true">
                    </div>
                    <button type="submit" class="btn btn-primary">Submit</button>
                  </form>

    {% for q in questions %}
    <div class="card">


        <div class="card-body">
            <h5 class="card-title">{{q.questions}}</h5>

        </div>

    </div>
    {% endfor %}

            </div>
            <div class="card mb-4" style="margin-right: -50px;">
                <div class="card-header">
                    Profile
                </div>
                <div class="card-body">
                    <h5 class="card-title">Welcome, {{user.username}}!</h5>
                    <a class="btn btn-md btn-primary" href="{%url 'signout' %}">Sign out</a>

                </div>

            </div>
        </div>

        <!-- Bootstrap core JavaScript
Hyper Text Markup Language file   length : 6,674   lines : 170   Ln : 155   Col : 1   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS
```

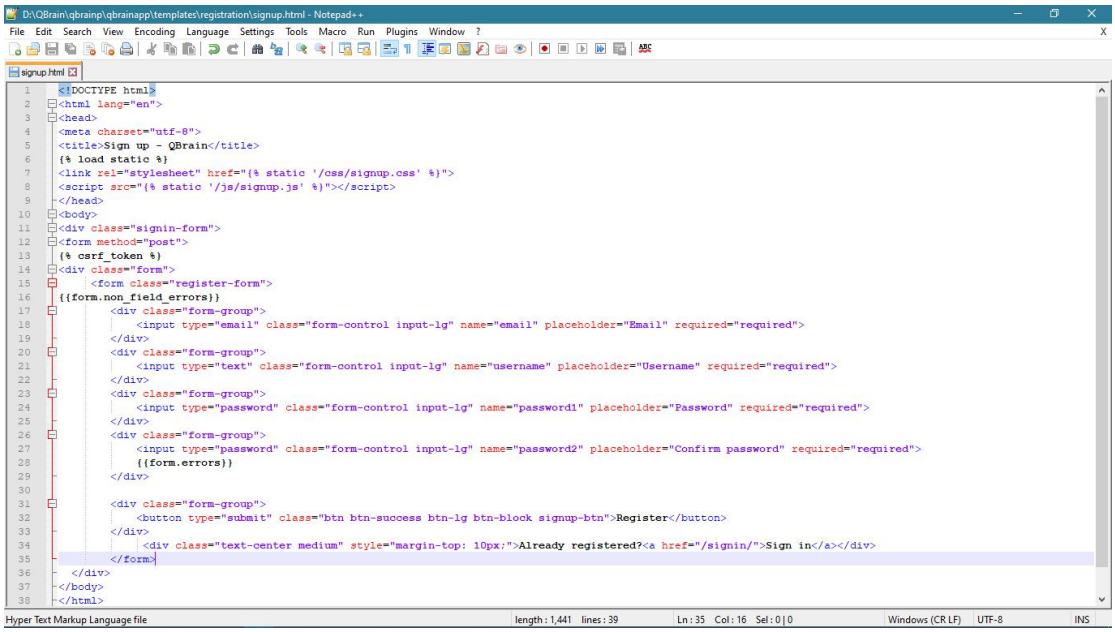


```
*D:\QBrain\qbrainp\qbrainapp\templates\qbrain\home.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
home.html
        <div class="card-body">
            <h5 class="card-title">{{q.questions}}</h5>

        </div>

    </div>
    {% endfor %}

            </div>
            <div class="card mb-4" style="margin-right: -50px;">
                <div class="card-header">
                    Profile
                </div>
                <div class="card-body">
                    <h5 class="card-title">Welcome, {{user.username}}!</h5>
                    <a class="btn btn-md btn-primary" href="{%url 'signout' %}">Sign out</a>

                </div>

            </div>
        </div>

        <!-- Bootstrap core JavaScript
    ================================================== -->
        <!-- Placed at the end of the document so the pages load faster -->
        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
            integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
            crossorigin="anonymous"></script>
        <script>window.jQuery || document.write('<script src="../../assets/js/vendor/jquery-slim.min.js"><\/script>')</script>
        <script src="../../assets/js/vendor/popper.min.js"></script>
        <script src="../../dist/js/bootstrap.min.js"></script>
        <!-- Just to make our placeholder images work. Don't actually copy the next line! -->
        <script src="../../assets/js/vendor/holder.min.js"></script>
</body>
 {% endblock %}
</html>
Hyper Text Markup Language file   length : 6,674   lines : 170   Ln : 170   Col : 1   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS
```

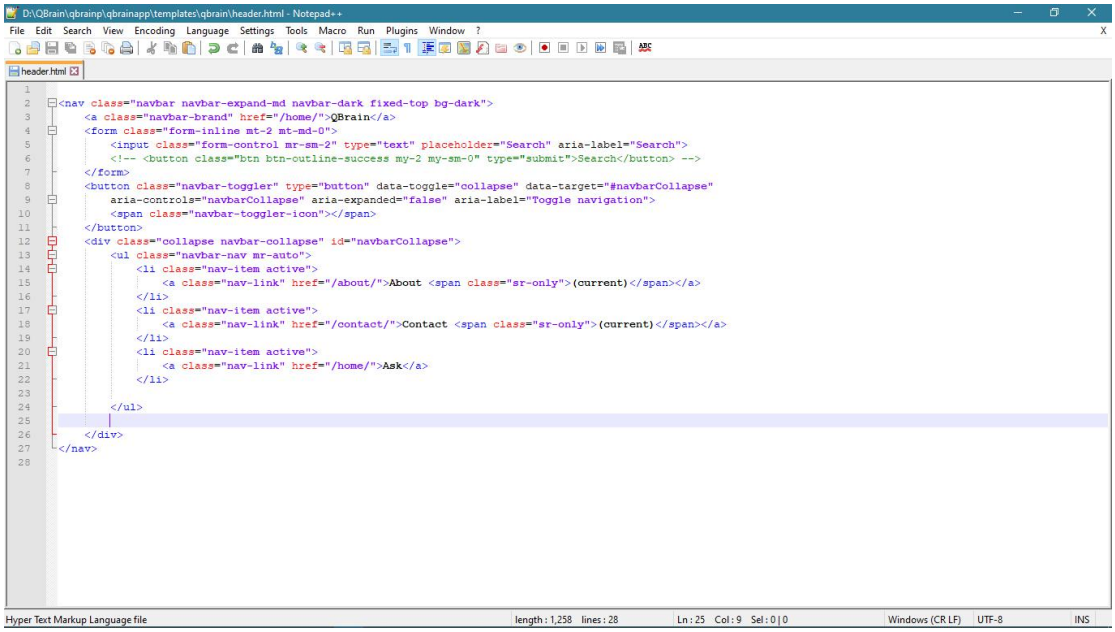
SIGNIN.HTML



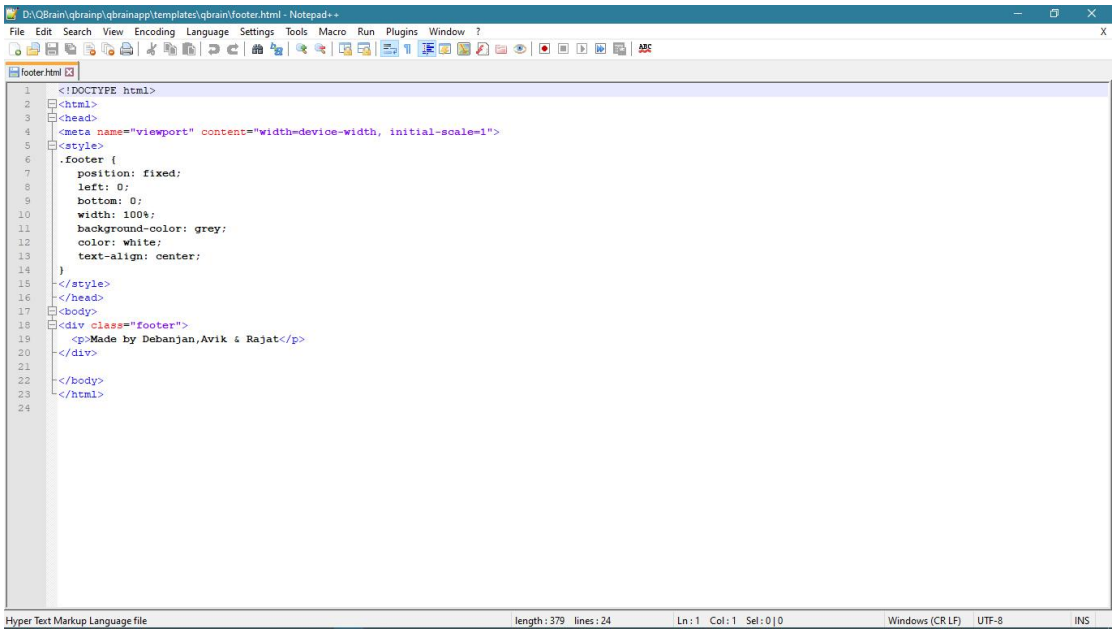
SIGNUP.HTML



HEADER.HTML



FOOTER.HTML



CONCLUSION

This website can be improved in future by many aspects. Each question can have a seperate webpage and each user with a their own profile page where all there questions and those question they answered will be there. The website can be launched globally by taking a proper privacy policy and copyright and purchasing a domain/host name.