# PROJECT REPORT FOR ‘DISCOPY’

| PROJECT SUBMITTED BY : | TRAINER NAME : |
|---|---|
| 1. DIKSHA RAJ<br>2. SHASHVAT PANDEY<br>3. ANISH KALRA<br>4. NIKHIL SAH<br>5. SUSHMITA SARRAF<br>6. SHARDA JHA | Venkatasubramanian R |

## 1. WHY DID WE CHOOSE THIS PROJECT?

Because we wanted to something for the student and give them a user friendly platform for academic purposes.

## 2. PROCESS REQUIREMENT

### HTML

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

### CSS

CSS is the language for describing the presentation of Web pages, including colors, lawet, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.

### Django

Django is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a 501(c) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspectionand configured via admin models.

## Features

- Contains development server and debugger
- RESTful request dispatching
- Uses multiple API
- Support for secure cookies (client side sessions)
- 100% WSGI 1.0 compliant
- Extensive documentation
- Extensions available to enhance features desired

## BOOTSTRAP

Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps.

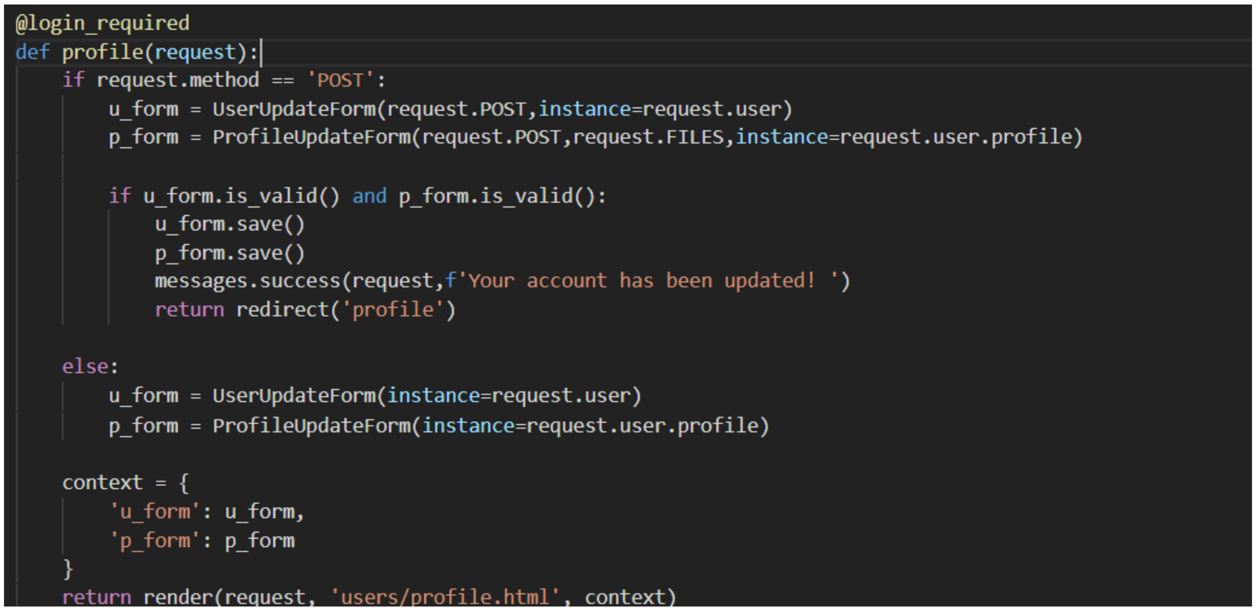
Responsive design makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display accordingly; the mobile first approach assumes that smartphones, tablets and task-specific mobile apps are employees' primary tools for getting work done and addresses the requirements of those technologies in design.

Bootstrap includes user interface components, lawets and JS tools along with the framework for implementation. The software is available precompiled or as source code.

### RESTful web service

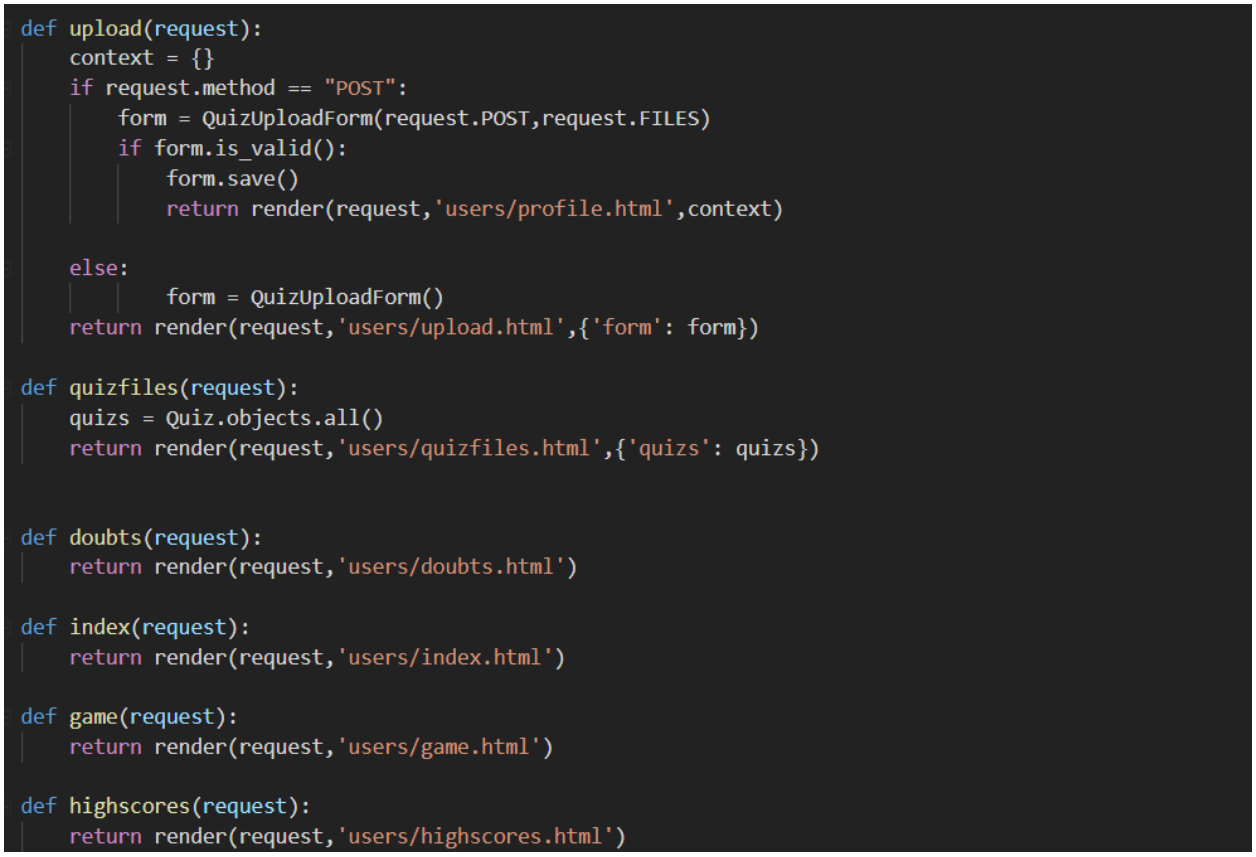
The REST architecture was originally designed to fit the HTTP protocol that the World Wide Web uses. Central to the concept of RESTful web services is the notion of resources. Resources are represented by URIs. The clients send requests to these URIs using the methods defined by the HTTP protocol, and possibly as a result of that the state of the affected resource changes.

## CODE AND FUNCTIONALITIES:



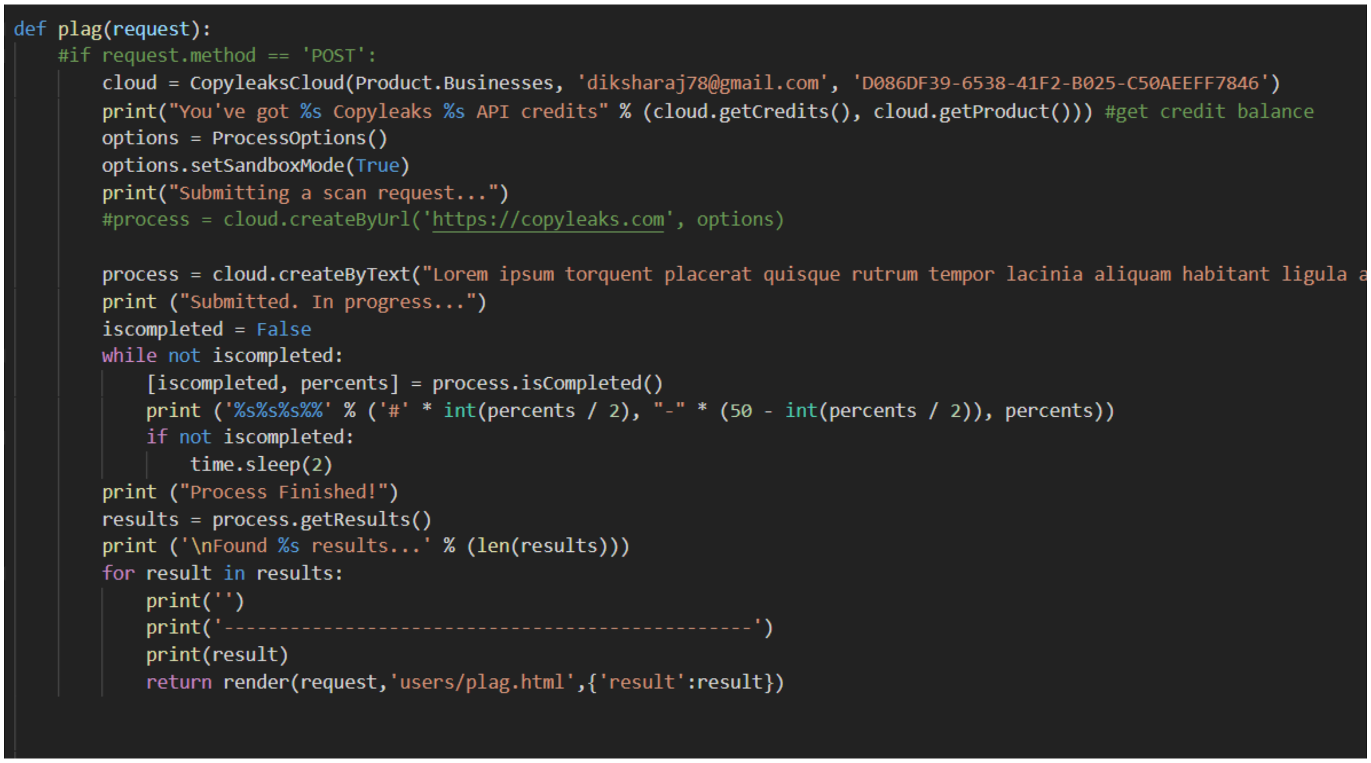
```python
@login_required
def profile(request):
    if request.method == 'POST':
        u_form = UserUpdateForm(request.POST,instance=request.user)
        p_form = ProfileUpdateForm(request.POST,request.FILES,instance=request.user.profile)

        if u_form.is_valid() and p_form.is_valid():
            u_form.save()
            p_form.save()
            messages.success(request,f'Your account has been updated! ')
            return redirect('profile')

    else:
        u_form = UserUpdateForm(instance=request.user)
        p_form = ProfileUpdateForm(instance=request.user.profile)

    context = {
        'u_form': u_form,
        'p_form': p_form
    }
    return render(request, 'users/profile.html', context)
```

This will display the users profile, which in turn would direct them to various other options. This will create a route to the login page/method and it will cross check the correct user name and password with the values inserted in the form.



```
def upload(request):
    context = {}
    if request.method == "POST":
        form = QuizUploadForm(request.POST,request.FILES)
        if form.is_valid():
            form.save()
            return render(request,'users/profile.html',context)

    else:
            form = QuizUploadForm()
    return render(request,'users/upload.html',{'form': form})

def quizfiles(request):
    quizs = Quiz.objects.all()
    return render(request,'users/quizfiles.html',{'quizs': quizs})


def doubts(request):
    return render(request,'users/doubts.html')

def index(request):
    return render(request,'users/index.html')

def game(request):
    return render(request,'users/game.html')

def highscores(request):
    return render(request,'users/highscores.html')
```

These functions would define where exactly the users would be directed if they went on to select various options.



```
def plag(request):
    #if request.method == 'POST':
        cloud = CopyleaksCloud(Product.Businesses, 'diksharaj78@gmail.com', 'D086DF39-6538-41F2-B025-C50AEEFF7846')
        print("You've got %s Copyleaks %s API credits" % (cloud.getCredits(), cloud.getProduct())) #get credit balance
        options = ProcessOptions()
        options.setSandboxMode(True)
        print("Submitting a scan request...")
        #process = cloud.createByUrl('https://copyleaks.com', options)

        process = cloud.createByText("Lorem ipsum torquent placerat quisque rutrum tempor lacinia aliquam habitant ligula a
        print ("Submitted. In progress...")
        iscompleted = False
        while not iscompleted:
            [iscompleted, percents] = process.isCompleted()
            print ('%s%s%s%%' % ('#' * int(percents / 2), "-" * (50 - int(percents / 2)), percents))
            if not iscompleted:
                time.sleep(2)
        print ("Process Finished!")
        results = process.getResults()
        print ('\nFound %s results...' % (len(results)))
        for result in results:
            print('')
            print('------------------------------------------------')
            print(result)
            return render(request,'users/plag.html',{'result':result})
```

This is the code to trigger the Copyleaks API which in turn would check users' submissions for plagiarism. The returned response is displayed on a separate HTML page.

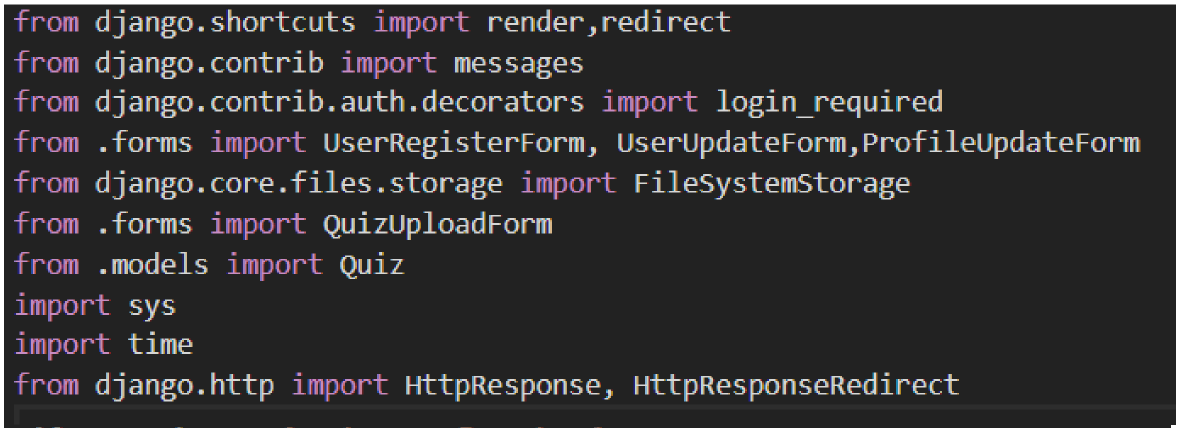
**Plagiarism API request Processing**

```
@app.route('/plagiarism/<string:id>', methods=['POST'])
def plagiarism(id):
# Create cursor
cur = mysql.connection.cursor()
# Execute
cur.execute("SELECT body FROM articles WHERE id = %s", [id])
if __name__ == '__main__':
cloud = CopyleaksCloud(Product.Education, 'shashvat233@gmail.com',
        '82ae5bde-ff2a-4481-9b88-2a413f65d355')
print("You've got %s Copyleaks %s API credits" % (cloud.getCredits(),
cloud.getProduct())) #get credit balance
options = ProcessOptions()
options.setSandboxMode(True)
print("Submitting a scan request…")
process = cloud.createByUrl('https://copyleaks.com', options)
# process = cloud.createByOcr('ocr-example.jpg', eOcrLanguage.English,
options)
# process = cloud.createByFile('test.txt', options)
```

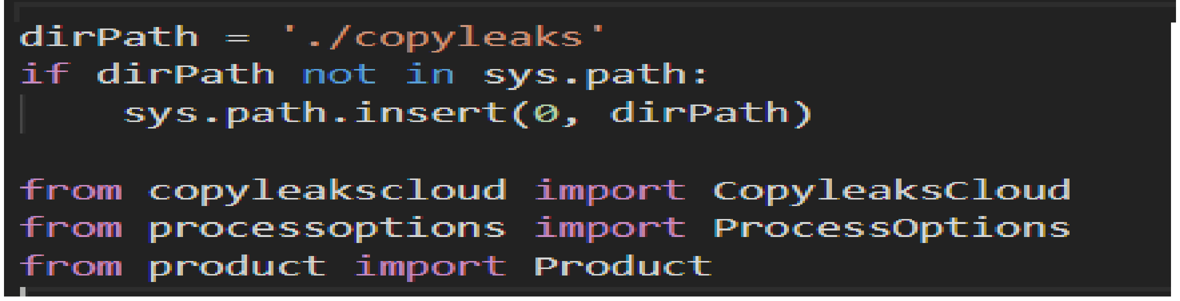
```
process = cloud.createByText("Lorem ipsum torquent placerat quisque
rutrum tempor lacinia aliquam habitant ligula arcu faucibus gravida,
aenean orci lacinia mattis purus consectetur conubia mauris amet nibh
consequat turpis dictumst hac ut nullam sodales nunc aenean pharetra,
aenean ut sagittis leo massa nisi duis nullam iaculis, nulla ultrices
consectetur facilisis curabitur scelerisque quisque primis elit
sagittis dictum felis ornare class porta rhoncus lobortis donec
praesent curabitur cubilia nec eleifend fringilla fusce vivamus
elementum semper nisi conubia dolor, eros habitant nisl suspendisse
venenatis interdum nulla interdum, libero urna maecenas potenti nam
habitant aliquam donec class sem hendrerit tempus.")
# processes, errors = cloud.createByFiles(['path/to/file1',
'path/to/file2'], options)
print ("Submitted. In progress…")
iscompleted = False
while not iscompleted:
# Get process status
[iscompleted, percents] = process.isCompleted()
print ('%s%s%s%%' % ('#' * int(percents / 2), "-" * (50 - int(percents
/ 2)), percents))
if not iscompleted:
time.sleep(2)
print ("Process Finished!")
# Get the scan results
results = process.getResults()
print ('\nFound %s results...' % (len(results)))
for result in results:
print('')
print('-------------------------------------------------')
print(result)
return str(result)
```

Here we are trying to send post request to the API in text format , where it is checking what percentage of the article is plagiarized and what sources on the internet are similar to the submitted article.

**LIBRARIES USED:**

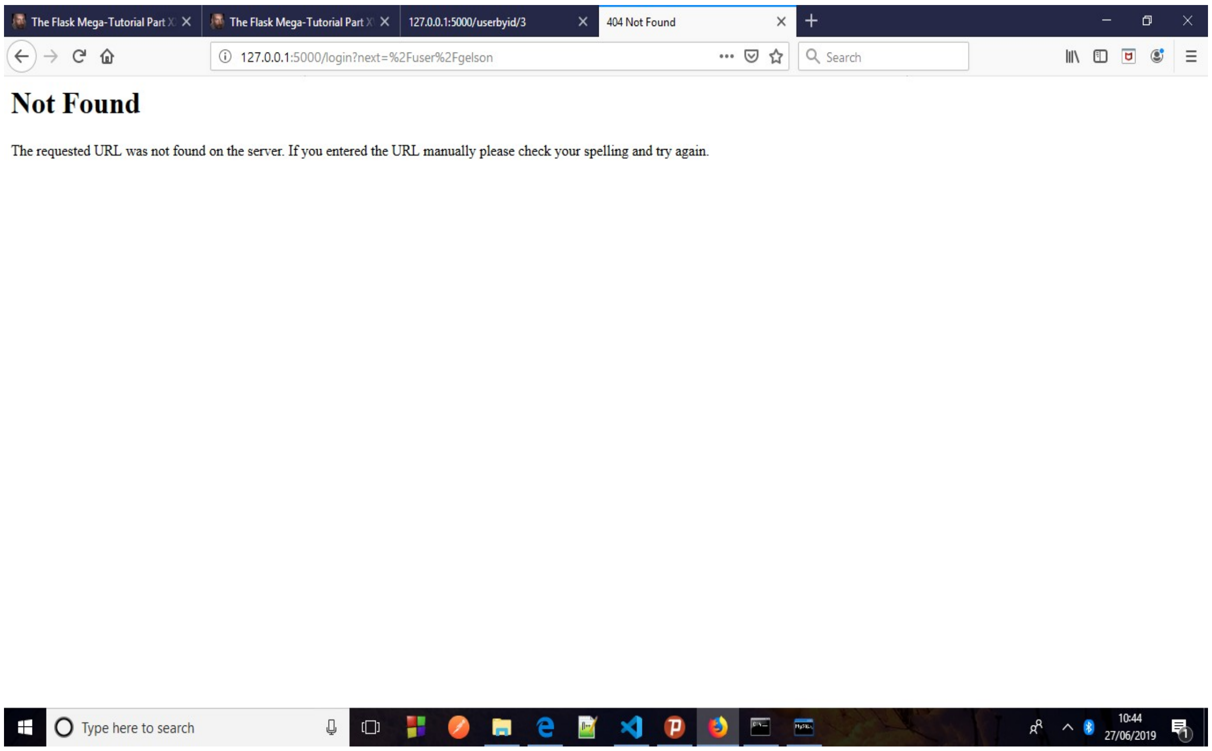


```
from django.shortcuts import render,redirect
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from .forms import UserRegisterForm, UserUpdateForm,ProfileUpdateForm
from django.core.files.storage import FileSystemStorage
from .forms import QuizUploadForm
from .models import Quiz
import sys
import time
from django.http import HttpResponse, HttpResponseRedirect
```



```
dirPath = './copyleaks'
if dirPath not in sys.path:
    sys.path.insert(0, dirPath)

from copyleakscloud import CopyleaksCloud
from processoptions import ProcessOptions
from product import Product
```

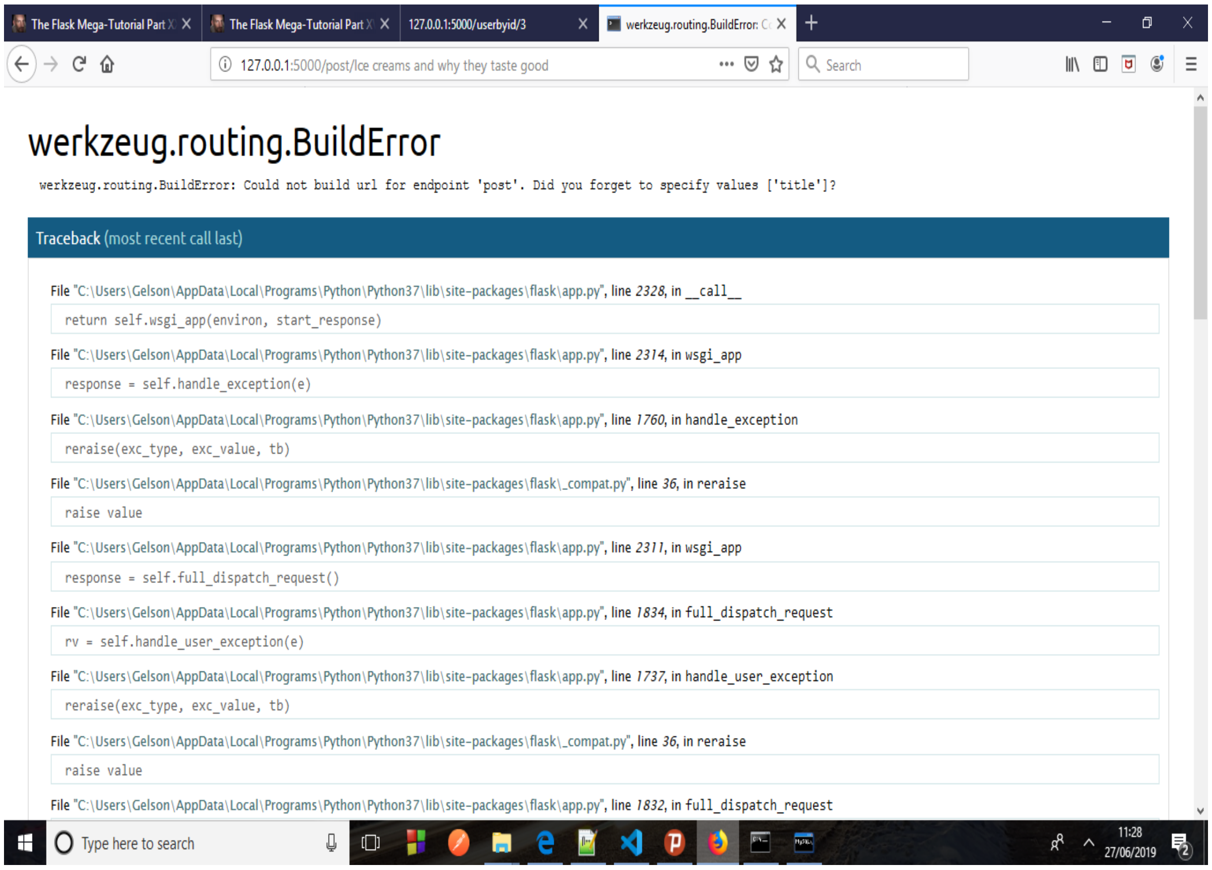
## TEST ERROR AND ANALYSIS


The Flask Mega-Tutorial Part
The Flask Mega-Tutorial Part
127.0.0.1:5000/userbyid/3
404 Not Found
127.0.0.1:5000/login?next=%2Fuser%2Fgelson
Search
Type here to search
10:44
27/06/2019

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

The HTTP 404 Not Found Error means that the webpage we were trying to reach could not be found on the server. It is a Client-side Error which means that either the page has been removed or moved and the URL was not changed accordingly, or that we typed in the URL incorrectly.

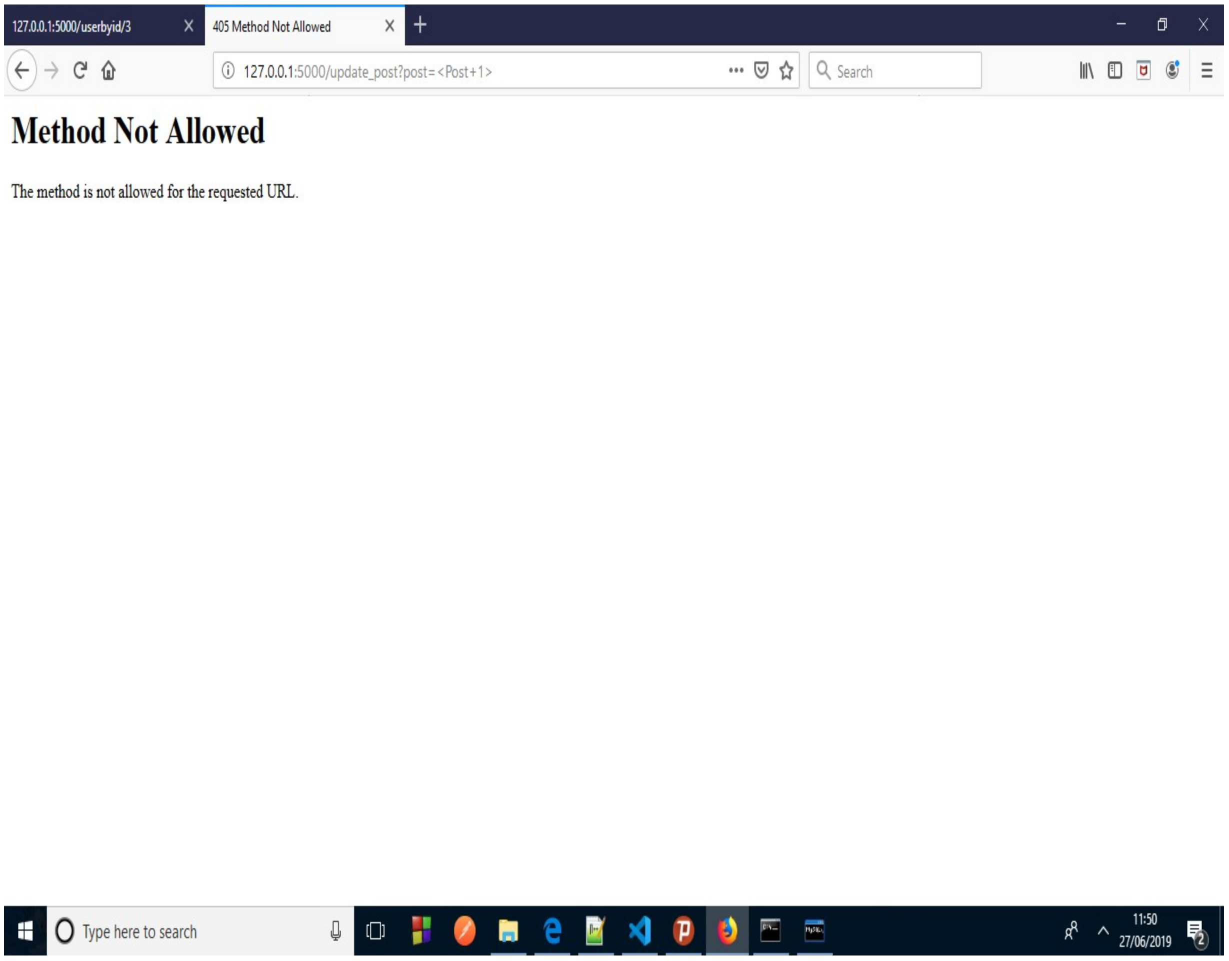

The Flask Mega-Tutorial Part
The Flask Mega-Tutorial Part
127.0.0.1:5000/userbyid/3
werkzeug.routing.BuildError: C
127.0.0.1:5000/post/Ice creams and why they taste good
Search
Type here to search
11:28
27/06/2019

werkzeug.routing.BuildError

```
werkzeug.routing.BuildError: Could not build url for endpoint 'post'. Did you forget to specify values ['title']?

Traceback (most recent call last)

File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 2328, in __call__
    return self.wsgi_app(environ, start_response)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 2314, in wsgi_app
    response = self.handle_exception(e)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 1760, in handle_exception
    reraise(exc_type, exc_value, tb)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\_compat.py", line 36, in reraise
    raise value
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 2311, in wsgi_app
    response = self.full_dispatch_request()
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 1834, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 1737, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\_compat.py", line 36, in reraise
    raise value
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 1832, in full_dispatch_request
```

When it comes to combining multiple controller or view we need a dispatcher. A simple way would be applying regular expression tests on PATH_INFO and call registered callback functions that return the value.

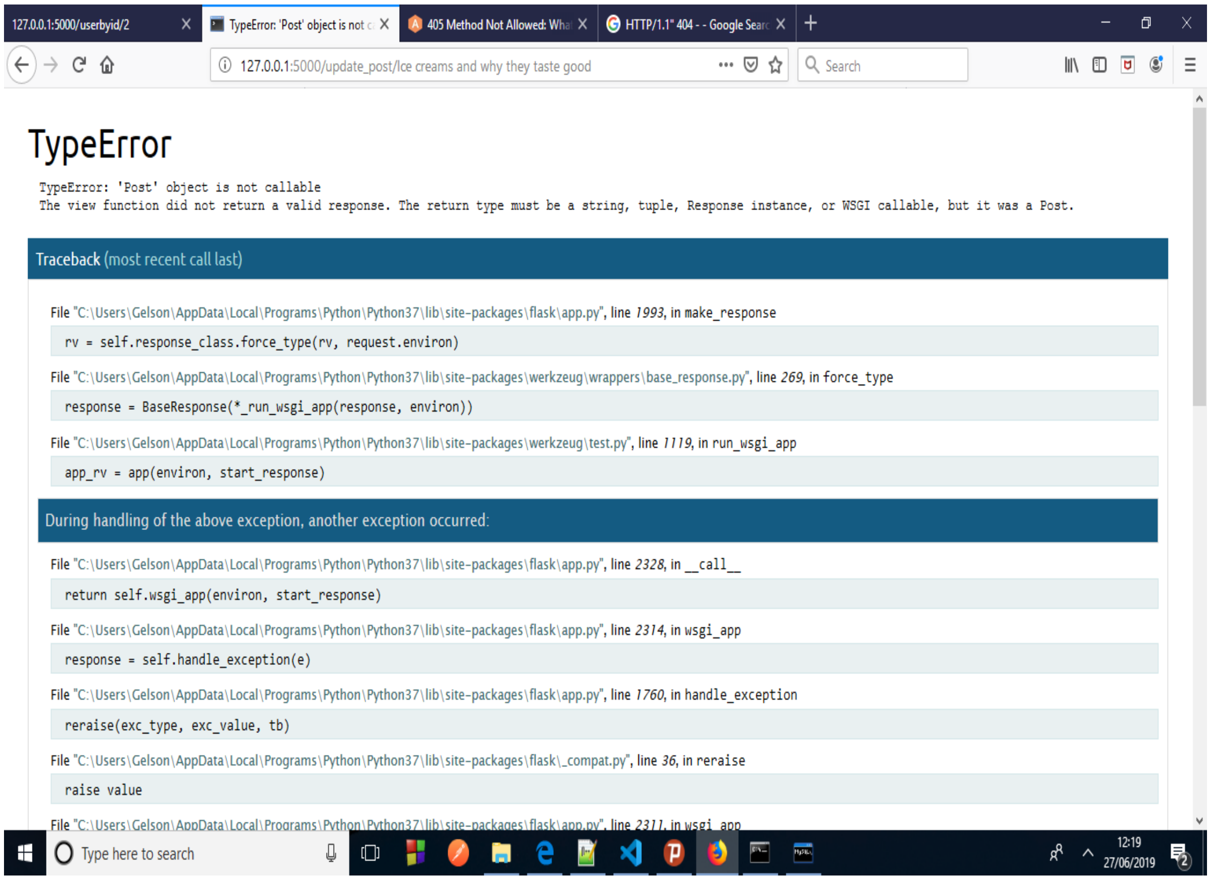
Werkzeug provides a much more powerful system, similar to Routes. All the objects mentioned on this page must be imported from **werkzeug.routing**, not from **werkzeug**!


127.0.0.1:5000/userbyid/3
405 Method Not Allowed
127.0.0.1:5000/update_post?post=<Post+1>

Method Not Allowed

The method is not allowed for the requested URL.

A **405 Method Not Allowed** Error is an HTTP response status code that indicates a web browser has requested access to one of web web pages and web web server received and recognized its HTTP **method**.



127.0.0.1:5000/update_post/Ice creams and why they taste good

TypeError

```
TypeError: 'Post' object is not callable
The view function did not return a valid response. The return type must be a string, tuple, Response instance, or WSGI callable, but it was a Post.
```

Traceback (most recent call last)

```
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 1993, in make_response
  rv = self.response_class.force_type(rv, request.environ)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\werkzeug\wrappers\base_response.py", line 269, in force_type
  response = BaseResponse(*_run_wsgi_app(response, environ))
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\werkzeug\test.py", line 1119, in run_wsgi_app
  app_rv = app(environ, start_response)
```

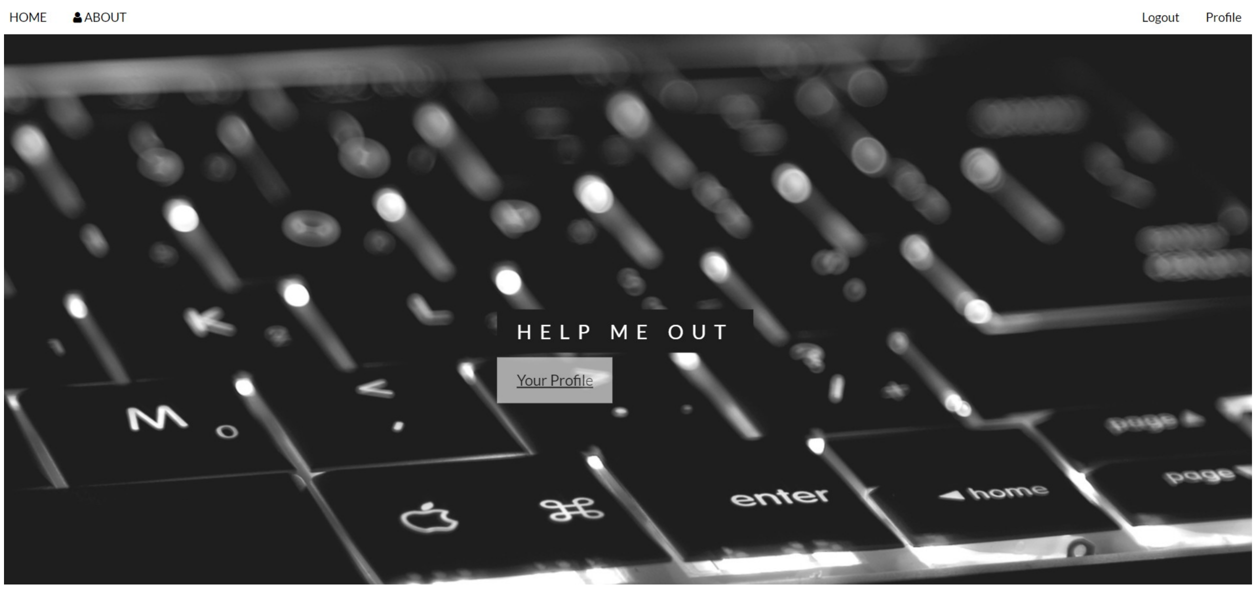
During handling of the above exception, another exception occurred:

```
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 2328, in __call__
  return self.wsgi_app(environ, start_response)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 2314, in wsgi_app
  response = self.handle_exception(e)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 1760, in handle_exception
  reraise(exc_type, exc_value, tb)
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\_compat.py", line 36, in reraise
  raise value
File "C:\Users\Gelson\AppData\Local\Programs\Python\Python37\lib\site-packages\flask\app.py", line 2311, in wsgi_app
```

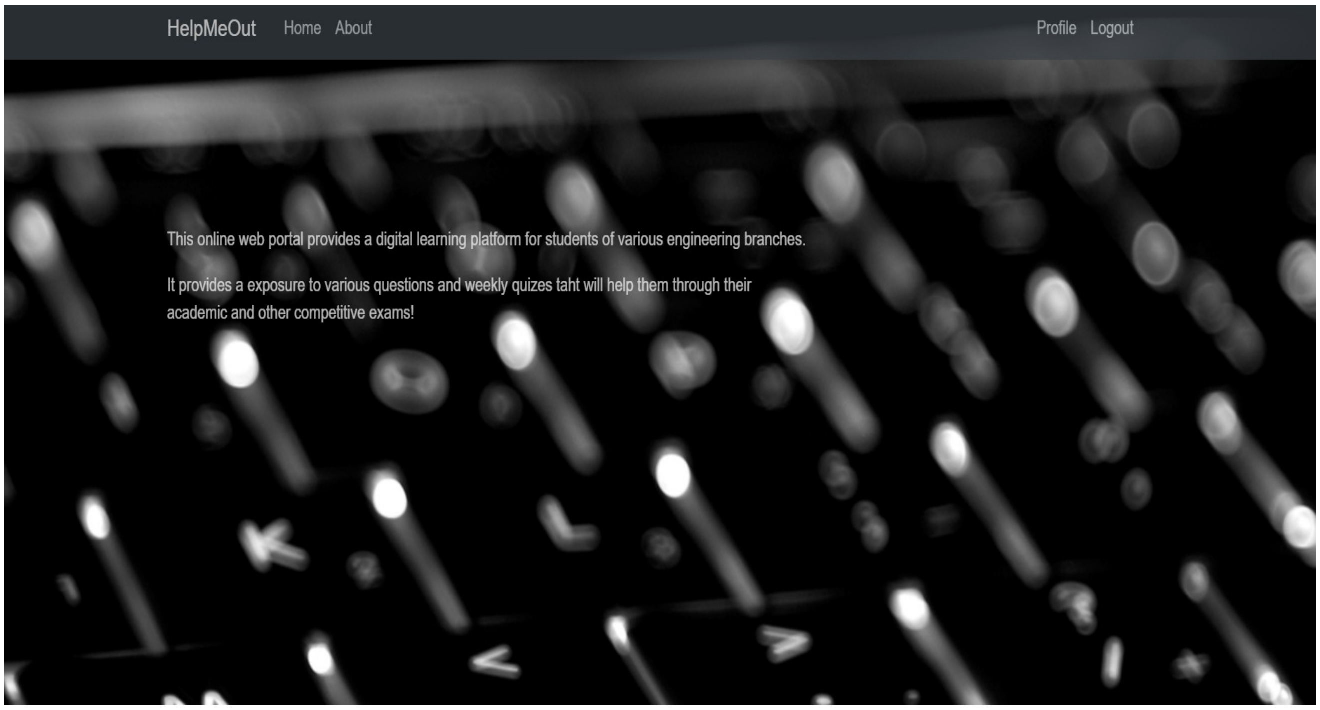
Raised when an operation or function is applied to an object of inappropriate type. The associated value is a string giving details about the type mismatch.

The point is we first created an object/instance of RegistrationForm using regsiter_form = RegistrationForm(), and then we tried regsiter_form(request.POST), so basically we are trying to again call an object/instance which is not allowed unless there is a __call__ method defined on wer class.

## TEST DATA



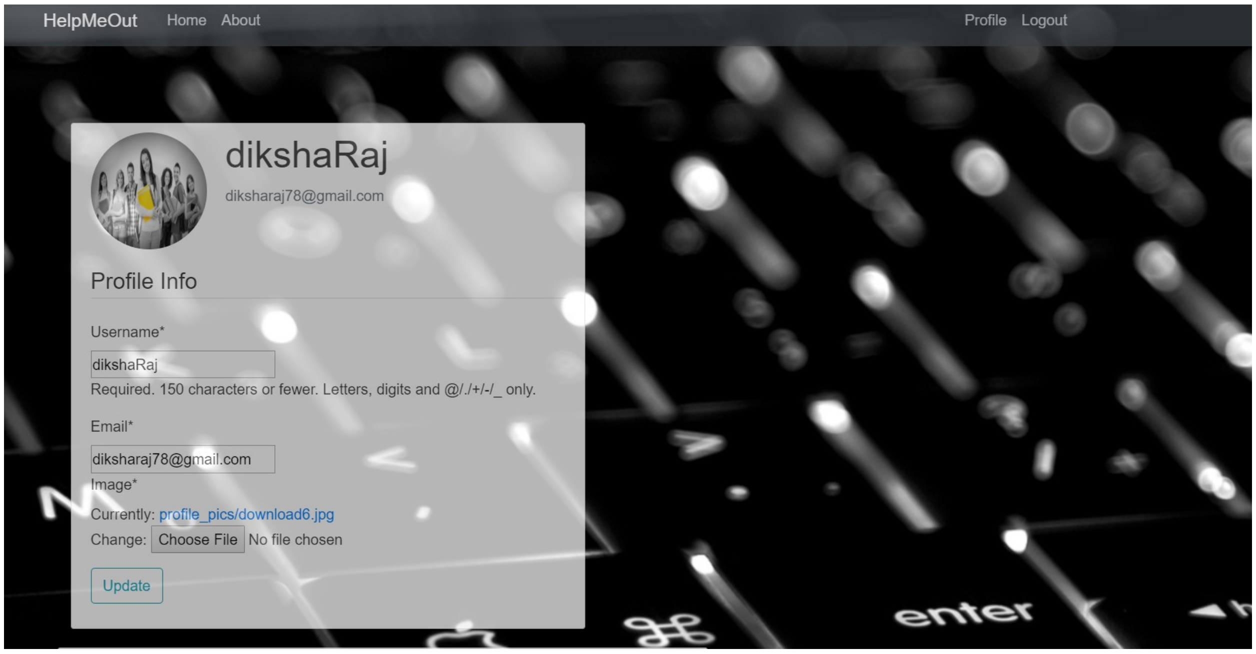
This is our home page.



This online web portal provides a digital learning platform for students of various engineering branches.

It provides a exposure to various questions and weekly quizes taht will help them through their academic and other competitive exams!

This is our about page.



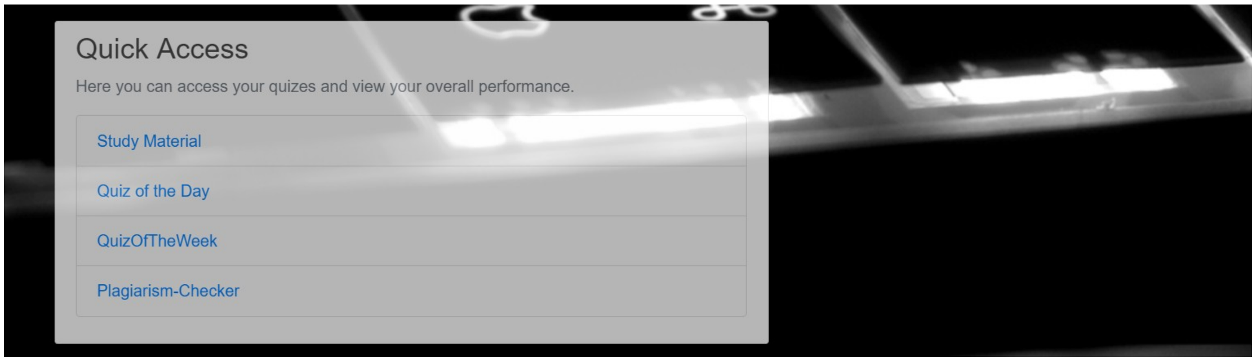
dikshaRaj
diksharaj78@gmail.com

Profile Info

Username*
dikshaRaj
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

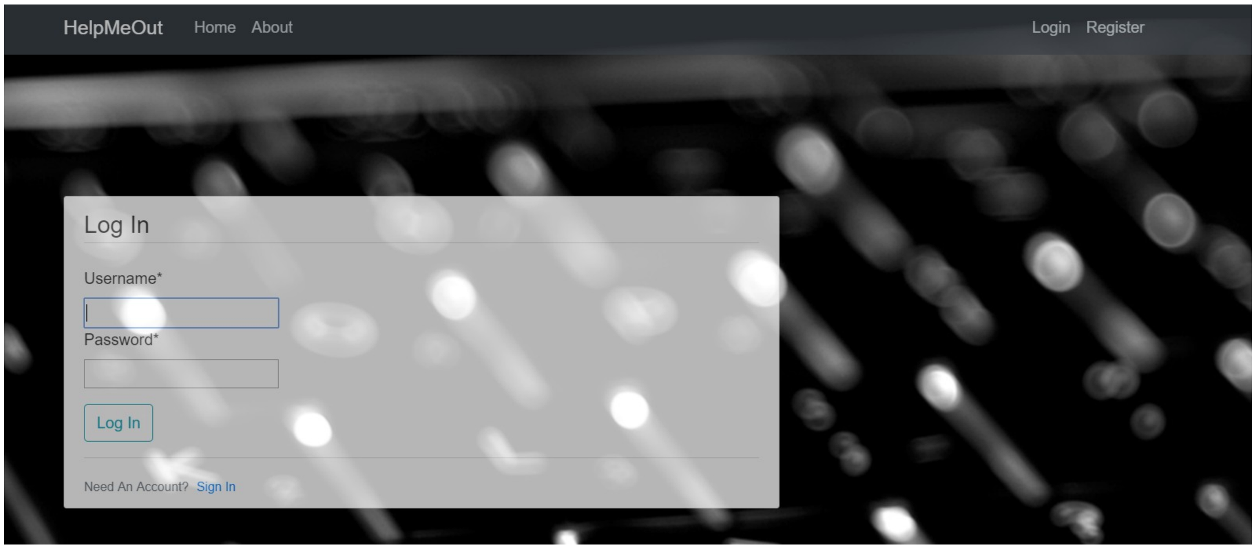
Email*
diksharaj78@gmail.com
Image*
Currently: profile_pics/download6.jpg
Change: Choose File No file chosen

This is the profile of the users.

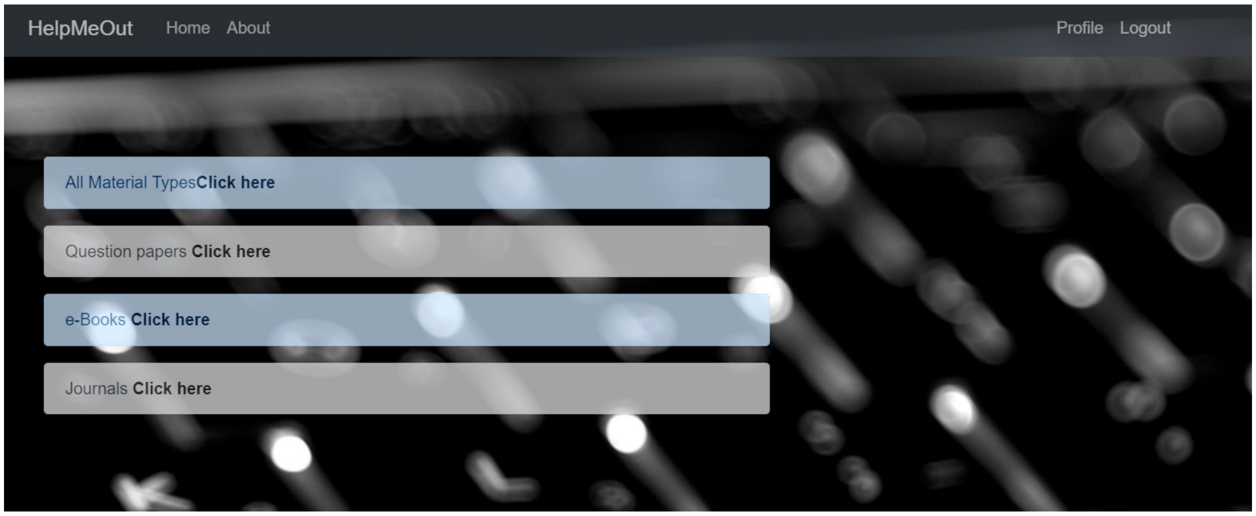

Quick Access

Here you can access your quizes and view your overall performance.

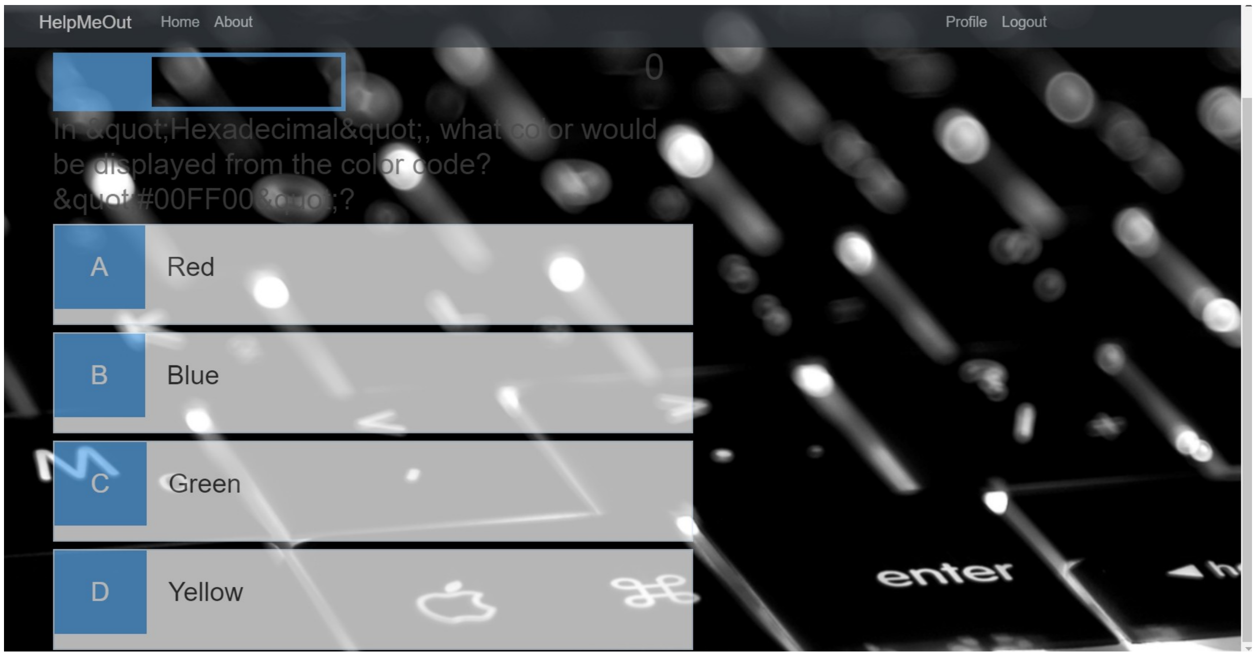
This provides quick access to the url.



From here the user can login.



Study Materials here.



In &quot;Hexadecimal&quot;, what color would be displayed from the color code? &quot;#00FF00&quot;?

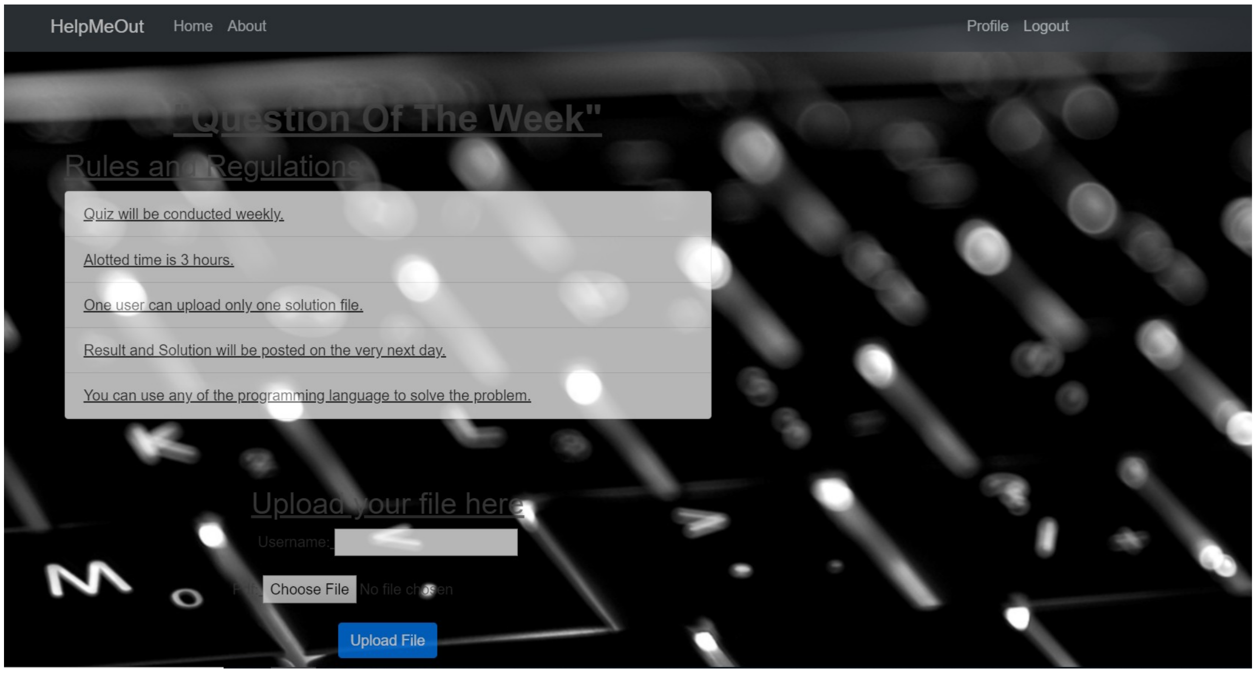
A Red

B Blue

C Green

D Yellow

Daily quiz on computer science

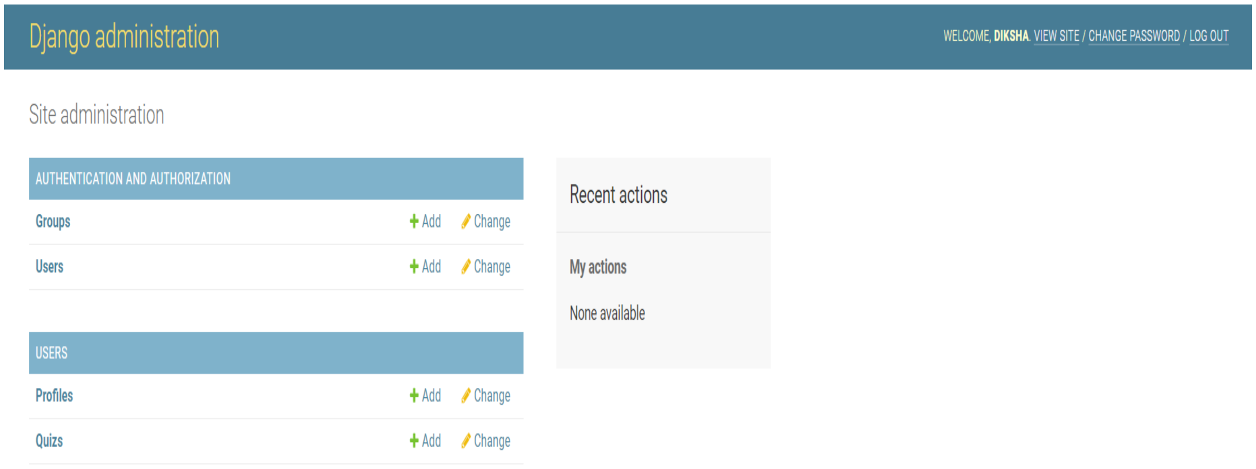


"Question Of The Week"

Rules and Regulations

- Quiz will be conducted weekly.
- Alotted time is 3 hours.
- One user can upload only one solution file.
- Result and Solution will be posted on the very next day.
- You can use any of the programming language to solve the problem.

Weekly quiz from where the users can upload their questions too.



Site administration

| AUTHENTICATION AND AUTHORIZATION | | |
|---|---|---|
| Groups | + Add | Change |
| Users | + Add | Change |

| USERS | | |
|---|---|---|
| Profiles | + Add | Change |
| Quizs | + Add | Change |

Recent actions

My actions

None available

He admin page stores all the data.

I certify that this software development document is complete, the unit __________ defined in this folder has successfully completed development and unit testing, and the unit is ready to be base lined and integrated into the system

Date_____________________ System Developer:_______________________

System Technical Lead Initials: _____________