Lab Manual

SUBJECT : OPEN SOURCE TECHNOLOGIES LAB

SUBJECT CODE : 515CAL02

SEMESTER : V SEMESTER

REGULATION : 2015

DEPARTMENT OF MCA

ADHIYAMAAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Dr. MGR NAGAR, HOSUR

CONTENTS

Ex. No.: 01 IMPLEMENT FILE/IMAGE UPLOADING USING PYTHON DJANGO

AIM:

To implement file/image uploading using python django.

PROCEDURE:

1. Start the program.
2. Import the file “from django import admin, forms, models and urls”.
3. Import view file “from django import render, requestContext & document”.
4. C:\Users\OSTLAB\Desktop>django-admin startproject upload

   These will create a folder named upload with following

   __init__.py, settings.py, urls.py, wsgi.py & manage.py
5. Navigate inside upload folder and run following command

   C:\Users\OSTLAB\PycharmProjects\upload>python manage.py startapp myapp

   This will create files named __init__.py, admin.py, models.py, tests.py, views.py & migrations Folder
6. Move myapp folder inside upload>upload> myapp
7. Modify files under upload folder.

8. Go to line in INSTALLED APPS

    8.1. under 'django.contrib.staticfiles', add the following lines 'upload.myapp'

9. Go to the line in MIDDLEWARE

    9.1 under ‘django.contrib.auth.middleware.AuthenticationMiddleware', add the following lines 'django.contrib.auth.middleware.SessionAuthenticationMiddleware',

10. Under TEMPLATES

    10.1 'DIRS': [os.path.join(BASE_DIR, 'upload', 'myapp', 'templates')

    10.2 'context_processors': [

        'django.contrib.auth.context_processors.auth',

        'django.template.context_processors.debug',

        'django.template.context_processors.i18n',

        'django.template.context_processors.media',

        'django.template.context_processors.static',

        'django.template.context_processors.tz',

        'django.contrib.messages.context_processors.messages',

    10.3 Delete Password Validation

    10.4 Under USE_TZ = True add the following

        MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

        MEDIA_URL = '/media/'

11. url.py

```
from django.conf.urls import include, url
from django.conf import settings
from django.conf.urls.static import static
from django.views.generic import RedirectView
from django.contrib import admin
urlpatterns = [
  url(r'^admin/', include(admin.site.urls)),
  url(r'^myapp/', include('upload.myapp.urls')),
  url(r'^$', RedirectView.as_view(url='/myapp/list/', permanent=True)),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

12. Modify contents of myapp folder files

12.1 admin.py

```
from django.contrib import admin
from upload.myapp.models import Document
admin.site.register(Document)
# Register your models here.
```

12.2 models.py

```
# -*- coding: utf-8 -*-
from django.db import models
```

```
class Document(models.Model):
    docfile = models.FileField(upload_to='documents/%Y/%m/%d')
```

12.3 create urls.py and paste it

```
# -*- coding: utf-8 -*-
from django.conf.urls import url
from upload.myapp.views import list
urlpatterns = [ url(r'^list/$', list, name='list') ]
```

12.4 views.py

```
# -*- coding: utf-8 -*-
from django.shortcuts import render
from django.template import RequestContext
from django.http import HttpResponseRedirect
from django.core.urlresolvers import reverse
from upload.myapp.models import Document
from upload.myapp.forms import DocumentForm
def list(request):
    # Handle file upload
    if request.method == 'POST':
```

```
    form = DocumentForm(request.POST, request.FILES)
    if form.is_valid():
    newdoc = Document(docfile=request.FILES['docfile'])
    newdoc.save()
# Redirect to the document list after POST
return HttpResponseRedirect(reverse('list'))
else:
form = DocumentForm()  # A empty, unbound form
# Load documents for the list page
documents = Document.objects.all()
# Render list page with the documents and the form
    return render (request, 'list.html', {'documents': documents, 'form': form})
```

12.5 create forms.py

```
# -*- coding: utf-8 -*-
from django import forms
class DocumentForm(forms.Form):
docfile = forms.FileField(label='Select a file')
```

13. Create a folder named templates and create a file named list.html

```
<!DOCTYPE html> <html> <head> <meta charset="utf-8">
<title>Minimal Django File Upload Example</title> </head>
```

```
<body> <!-- List of uploaded documents --> {% if documents %}
<ul> {% for document in documents %} <li>
<a href="{{ document.docfile.url }}">{{ document.docfile.name }}</a></li>
{% endfor %} </ul> {% else %} <p> No documents.</p>
{% endif %} <!-- Upload form. Note enctype attribute! -->
<form action="{% url "list" %}" method="post" enctype="multipart/form-data">
{% csrf_token %} <p>{{ form.non_field_errors }}</p>  <p> {
{form.docfile.label_tag }} {{ form.docfile.help_text }}</p> <p>
{{ form.docfile.errors }} {{ form.docfile }} </p> <p>
<input type="submit" value="Upload"/></p> </form> </body> </html>
```

14. Under Migrations folder create a file named 0001_initial.py and paste it

```
# -*- coding: utf-8 -*-
from __future__ import unicode_literals
from django.db import models, migrations
class Migration(migrations.Migration):
    dependencies = [    ]
    operations = [migrations.CreateModel(name='Document', fields=[
    ('id', models.AutoField(verbose_name='ID', serialize=False, auto_created=True,
    primary_key=True)), ('docfile', models.FileField (upload_to =b'documents
    /%Y/%m/%d')),]
```

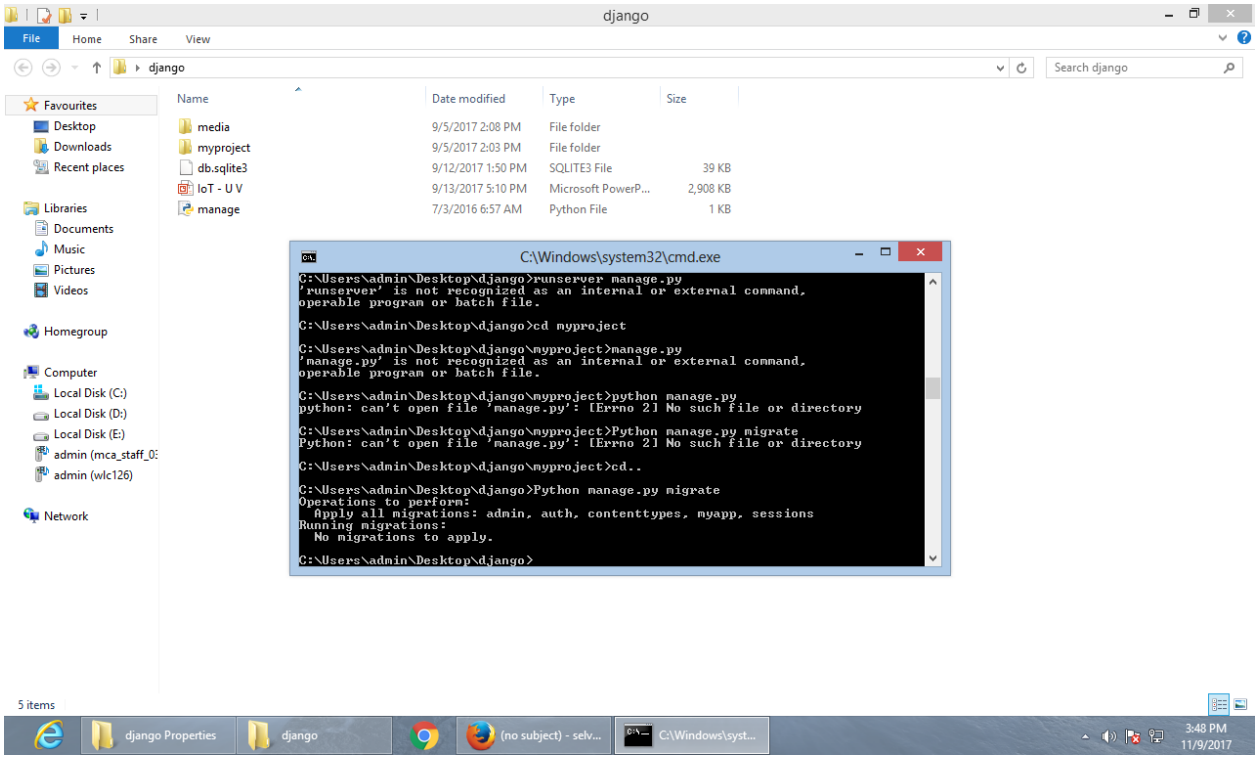
15. Run Python manage.py migrate

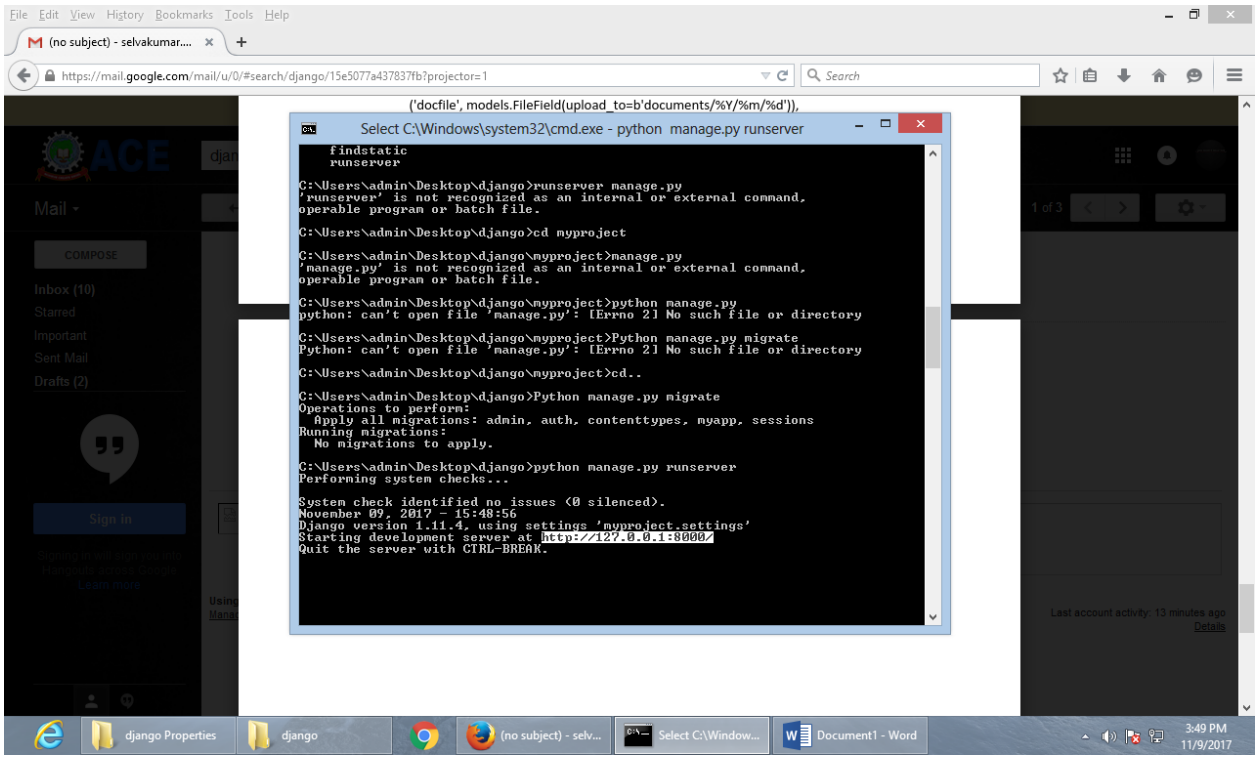
16. Run python manage.py runserver

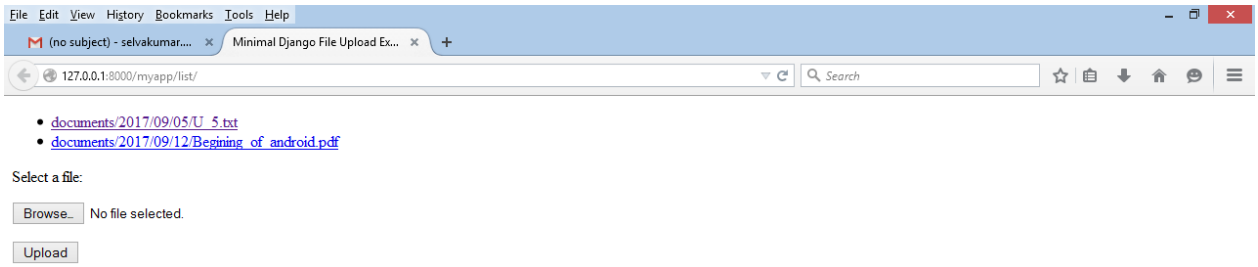
17. Open browser and type the url http://127.0.0.1:8000/

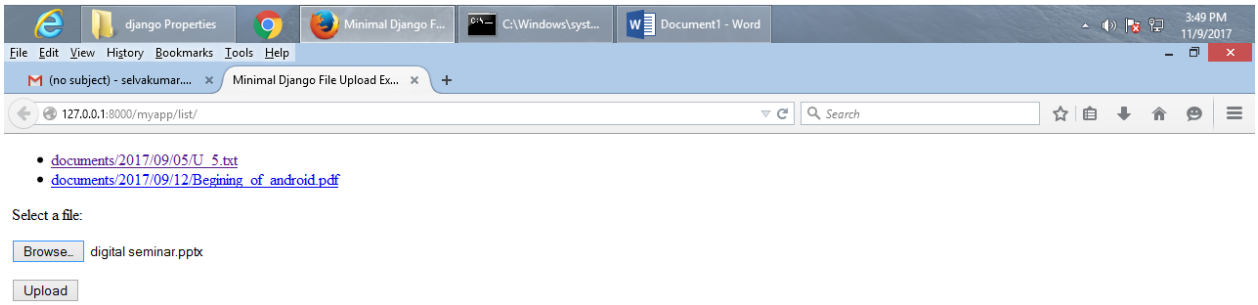
18. Upload a file and your upload files will be stored in media folder under your project folder.

IMPLEMENT FILE/IMAGE UPLOADING USING PYTHON DJANGO

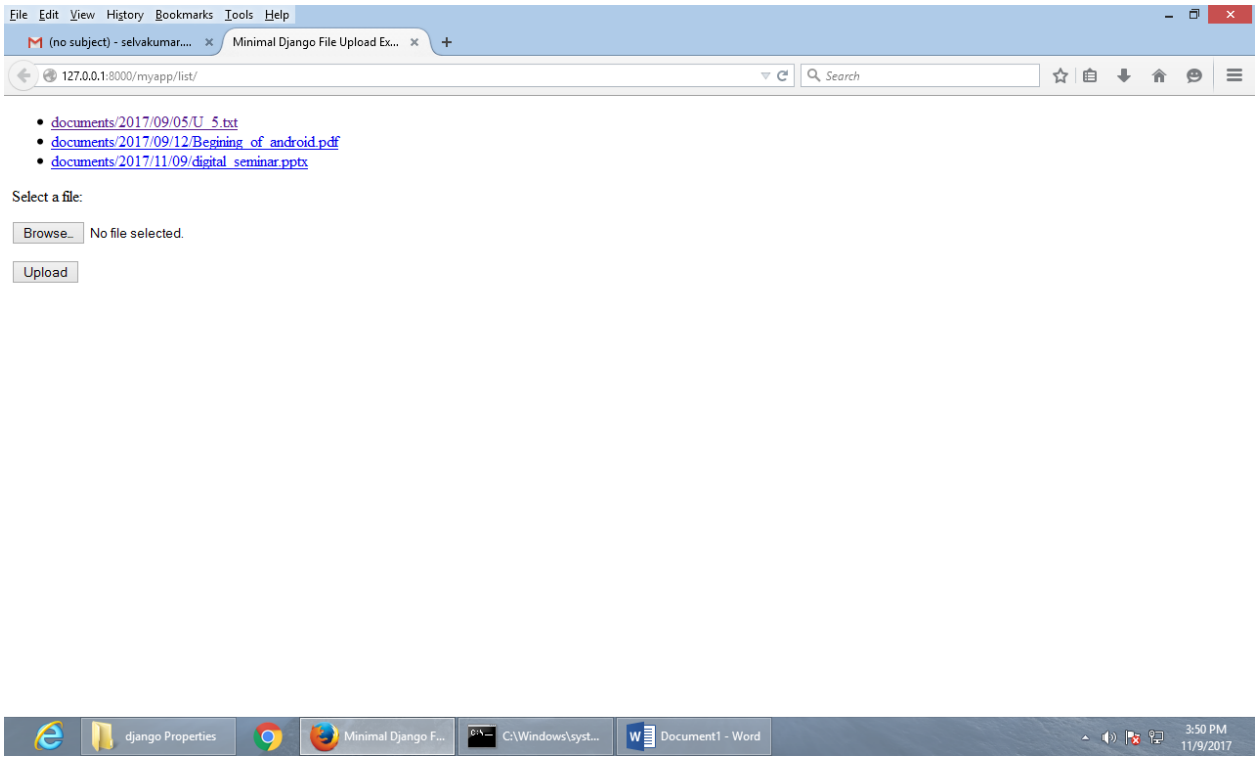





File Edit View History Bookmarks Tools Help
(no subject) - selvakumar....
Minimal Django File Upload Ex...
127.0.0.1:8000/myapp/list/
Search
documents/2017/09/05/U_5.txt
documents/2017/09/12/Begining_of_android.pdf
Select a file:
Browse...
No file selected.
Upload


django Properties
Minimal Django F...
C:\Windows\syst...
Document1 - Word
3:49 PM
11/9/2017
File Edit View History Bookmarks Tools Help
(no subject) - selvakumar....
Minimal Django File Upload Ex...
127.0.0.1:8000/myapp/list/
Search
documents/2017/09/05/U_5.txt
documents/2017/09/12/Begining_of_android.pdf
Select a file:
Browse...
digital seminar.pptx
Upload


django Properties
Minimal Django F...
C:\Windows\syst...
Document1 - Word
3:50 PM
11/9/2017


File Edit View History Bookmarks Tools Help
(no subject) - selvakumar....
Minimal Django File Upload Ex...
127.0.0.1:8000/myapp/list/
Search
documents/2017/09/05/U_5.txt
documents/2017/09/12/Begining_of_android.pdf
documents/2017/11/09/digital_seminar.pptx
Select a file:
Browse..
No file selected.
Upload
django Properties
Minimal Django F...
C:\Windows\syst...
Document1 - Word
3:50 PM
11/9/2017

Ex. No.: 02 DESIGN LOGIN PAGE USING PYTHON FLASK WEB APPLICATION

AIM:

To design login page using python flask.

PROCEDURE:

1. Start the program
2. Define the files “def.index (), def.login ()”.
3. Request the form to get a “Username & Password”.
4. Create an “app.run” method to get an HOST and PORT number.
5. Design a login form using “Text box and Login button” to get a values.
6. Run the file in command prompt to get a Host and Port number.
7. Stop the program.

CODING:

app.py

```
from flask import Flask, render_template, redirect, url_for, request
app = Flask(__name__)
@app.route('/')
def home():
```

```
        return "Hello, World!"
@app.route('/welcome')
def welcome():
        return render_template("welcome.html")
@app.route('/login', methods=['GET', 'POST'])
def login():
        error = None
        if request.method == 'POST':
        if request.form['username'] != 'admin' or request.form['password'] != 'admin':
                error = 'Invalid credentials. Please Try Again.'
        else:
                return redirect(url_for('home'))
return render_template('login.html', error=error)
if __name__ == '__main__':
        app.run(debug=True)
```

login.html

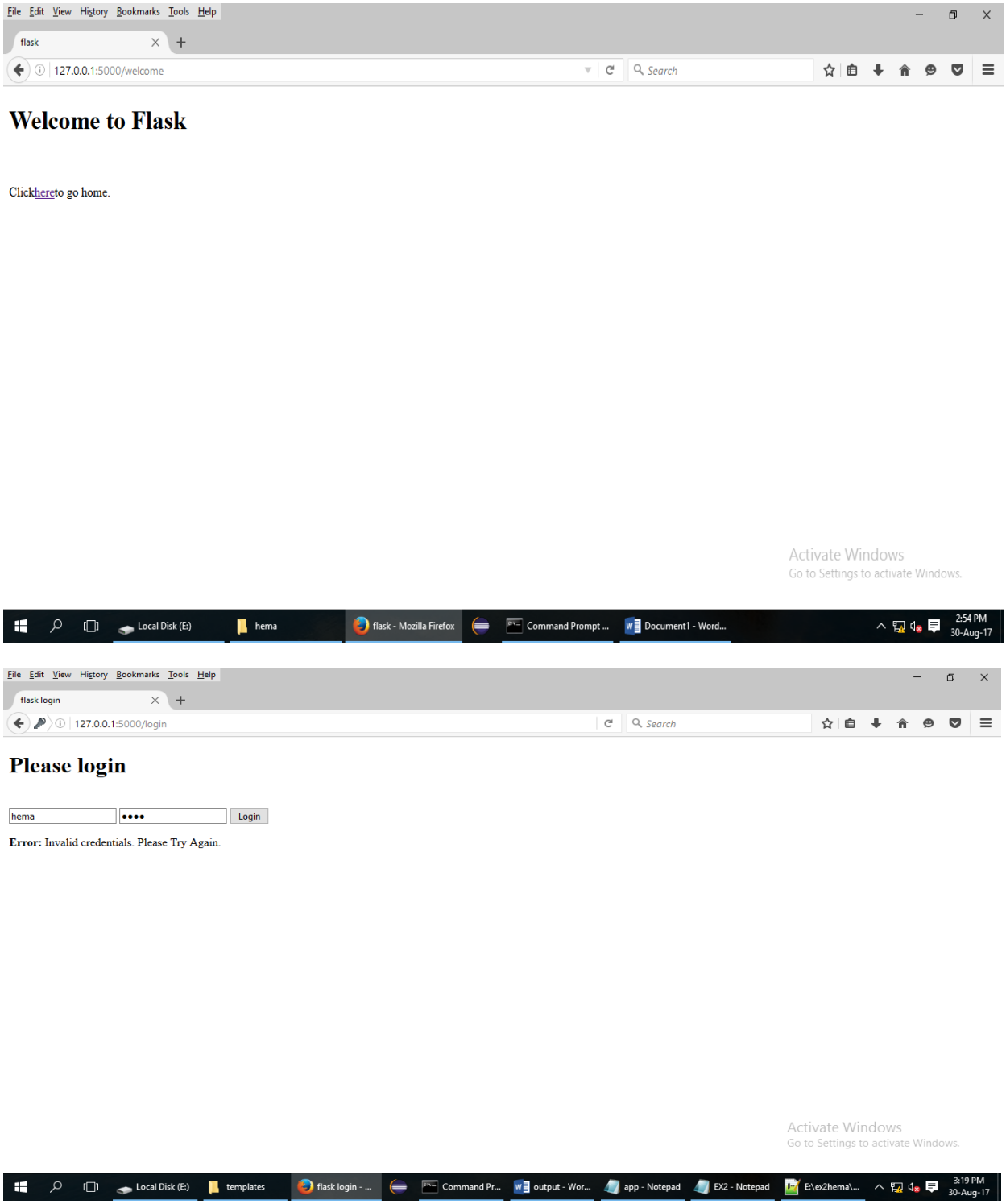
```
<!DOCTYPE html>
<html>
<head>
<title>flask login</title>
```

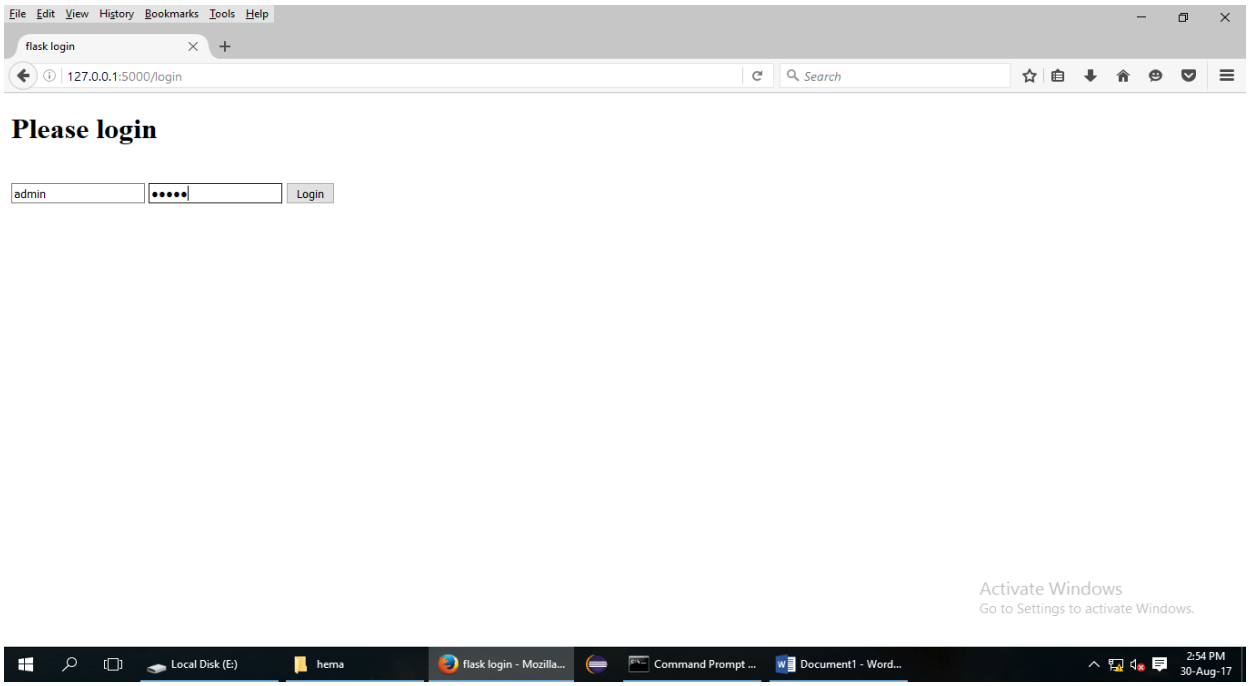
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="static/bootstrap.min.css" rel="stylesheet"  media="screen">

</head>

<body>

<div class="container">

<h1>Please login</h1>

<br>

<form action="" method="post">

<input type="text" placeholder="username" name="username" value="{{ request.form.username }}">

<input type="password" placeholder="Password" name="password" value="{{ request.form.password }}">

<input class="btn btn-default" type="submit" value="Login">

</form>

{% if error %}

<p class="error"><strong>Error:</strong> {{ error }}</p>

{% endif%}

</div>

</body>

</html>
```

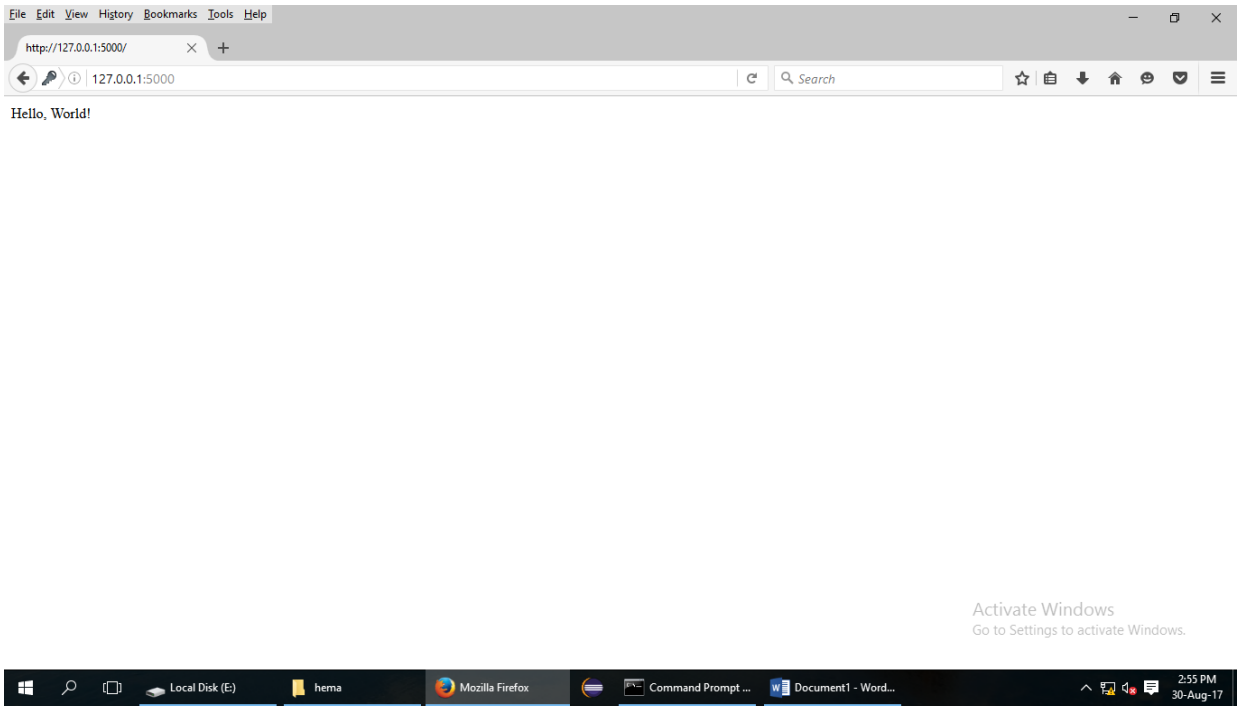
welcome.html

```
<!DOCTYPE html>

<html>

<head>

<title>flask</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="static/bootstrap.min.css" rel="stylesheet"  media="screen">

</head>

<body>

<div class="container">

<h1>Welcome to Flask</h2>

<br>

<p>Click<a href="/">here</a>to go home.</p>

</div>

</body>

</html>
```

DESIGN LOGIN PAGE USING PYTHON FLASK WEB APPLICATION




File Edit View History Bookmarks Tools Help
flask login
127.0.0.1:5000/login
Search
Please login
admin
Login
Activate Windows
Go to Settings to activate Windows.
Local Disk (E:)
hema
flask login - Mozilla...
Command Prompt ...
Document1 - Word...
2:54 PM
30-Aug-17


File Edit View History Bookmarks Tools Help
http://127.0.0.1:5000/
127.0.0.1:5000
Search
Hello, World!
Activate Windows
Go to Settings to activate Windows.
Local Disk (E:)
hema
Mozilla Firefox
Command Prompt ...
Document1 - Word...
2:55 PM
30-Aug-17

Ex. No.: 03 CREATE A MDI GUI COMPONENT USING PYTHON PYQT

AIM:

To create a MDI GUI component using python PyQt.

PROCEDURE:

1. Start the program
2. Import the file “from PyQt4 import QtGui”.
3. Create an class “MyDialog(QtGui.QDialog)”.
4. Define the file “def__init_(self,parent=none)”.
5. Design a window using “button box, text browser and addwidget to verticalLayout.
6. Create an another class “MyWindow(QtGui.QWidget)”
7. Define the “def_on_pushbutton_clicked(self)”.
8. Call the methods “MyWindow() and show()” to print a values.
9. Run and Stop the program.

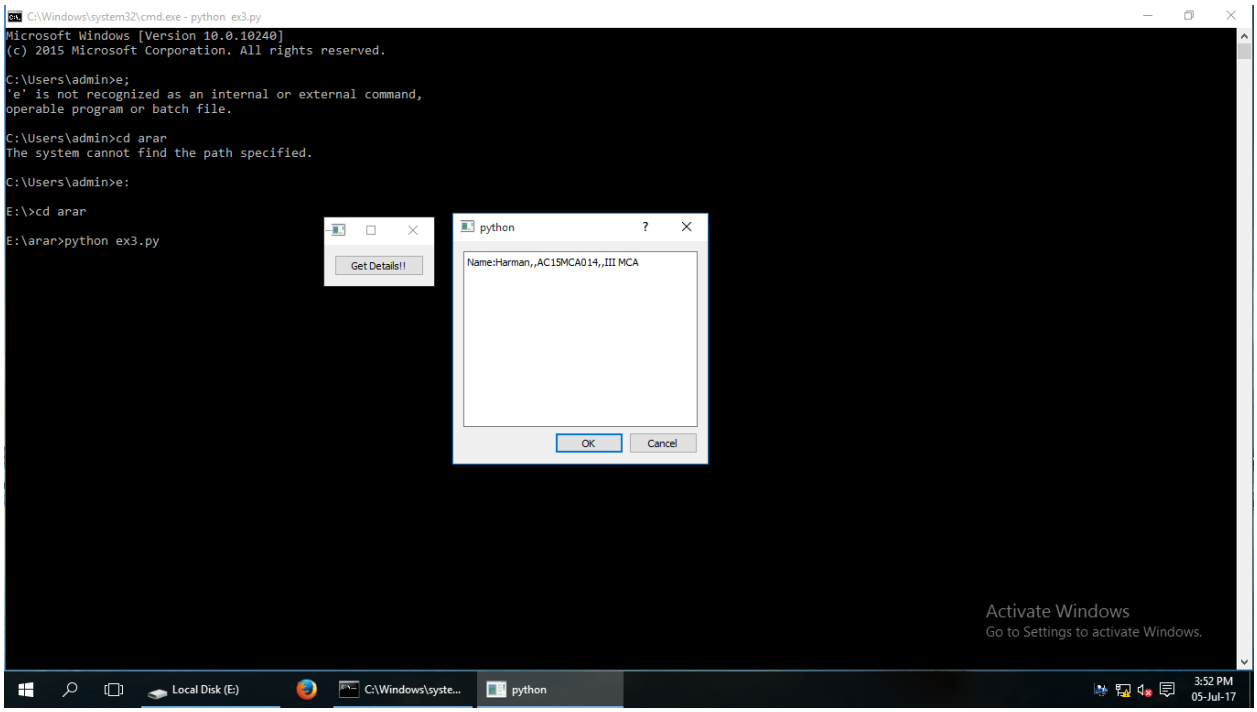
CODING:

```
from PyQt4 import QtCore, QtGui
class MyDialog(QtGui.QDialog):
   def __init__(self, parent=None):
```

```
        super(MyDialog, self).__init__(parent)
        self.buttonBox = QtGui.QDialogButtonBox(self)
        self.buttonBox.setOrientation(QtCore.Qt.Horizontal)
self.button Box.setStandardButtons (QtGui.QDialogButtonBox.Cancel
QtGui.QDialogButtonBox.Ok)
        self.textBrowser = QtGui.QTextBrowser(self)
        self.textBrowser.append("Name:Harman,,AC15MCA014,,III MCA")`
        self.verticalLayout = QtGui.QVBoxLayout(self)
        self.verticalLayout.addWidget(self.textBrowser)
        self.verticalLayout.addWidget(self.buttonBox)
  class MyWindow(QtGui.QWidget):
     def __init__(self, parent=None):
      super(MyWindow, self).__init__(parent)
     self.pushButtonWindow = QtGui.QPushButton(self)
            self.pushButtonWindow.setText("Get Details!!")
            self.pushButtonWindow.clicked.connect(self.on_pushButton_clicked)
     self.layout = QtGui.QHBoxLayout(self)
     self.layout.addWidget(self.pushButtonWindow)
     self.dialogTextBrowser = MyDialog(self)
     @QtCore.pyqtSlot()
```

```
 def on_pushButton_clicked(self):
 self.dialogTextBrowser.exec_()
if __name__ == "__main__":
  import sys
   app = QtGui.QApplication(sys.argv)
   app.setApplicationName('MyWindow')
   main = MyWindow()
   main.show()
   sys.exit(app.exec_())
```

CREATING A MDI,GUI COMPONENT USING PYTHON PYQT



Ex. No.: 04 CREATE A QMESSAGEBOX USING PYTHON LANGUAGE

AIM:

To create a QMessageBox using python language.

PROCEDURE:

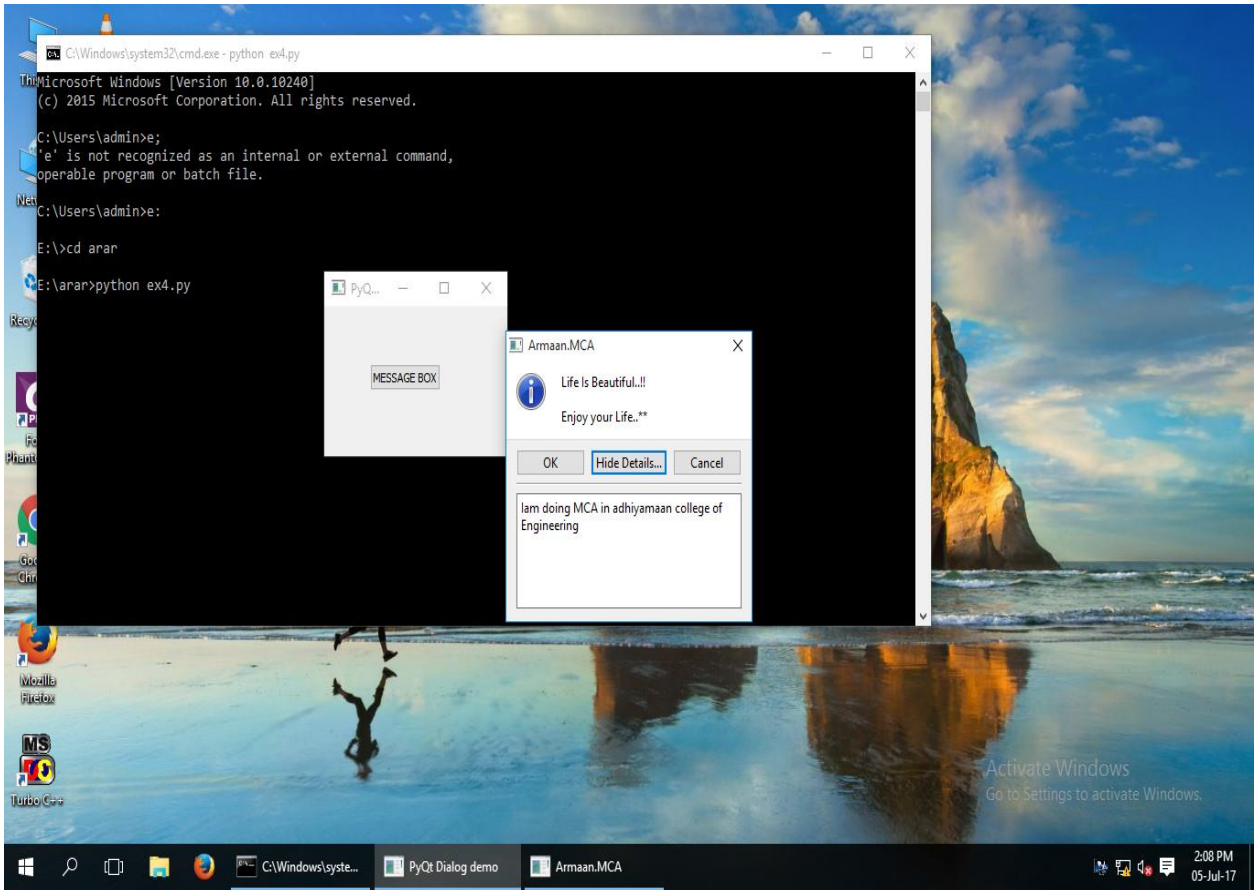
1. Start the program.
2. Import the file “from PyQt4.QtGui import* and from PyQt4.QtGui import*".
3. Define the “def.window()".
4. Design a window using “Qwidget, QPushbutton".
5. Define the “def.showdialog()".
6. Call an QMessageBox() method to print the values.
7. Define an “def.msgbtn()" to call the “Window()" method.
8. Stop the program.

CODING:

```
import sys
from PyQt4.QtGui import *
from PyQt4.QtCore import *
def window():
```

```
app = QApplication(sys.argv)

w = QWidget()        b = QPushButton(w)

b.setText("MESSAGE BOX")

b.move(50,50)        b.clicked.connect(showdialog)

w.setWindowTitle("PyQt Dialog demo")

w.show()      sys.exit(app.exec_())

def showdialog():     msg = QMessageBox()

msg.setIcon(QMessageBox.Information)

msg.setText("Life Is Beautiful..!!")

msg.setInformativeText("Enjoy your Life..**")

msg.setWindowTitle("Armaan.MCA")

msg.setDetailedText("Iam doing MCA in adhiyamaan college of Engineering")

msg.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

msg.buttonClicked.connect(msgbtn)

retval = msg.exec_()

print ("value of pressed message box button:", retval)

def msgbtn(S):

print( "Button pressed is:",i.text())

if __name__ == '__main__':

window()
```

CREATING A QMESSAGE BOX USING PYTHON LANGUAGE



Ex. No.: 05 DRAW AN API FOR GEOMETRIC SHAPES USING WXPYTHON

AIM:

To draw an API for geometric shapes using wxpython.

PROCEDURE:

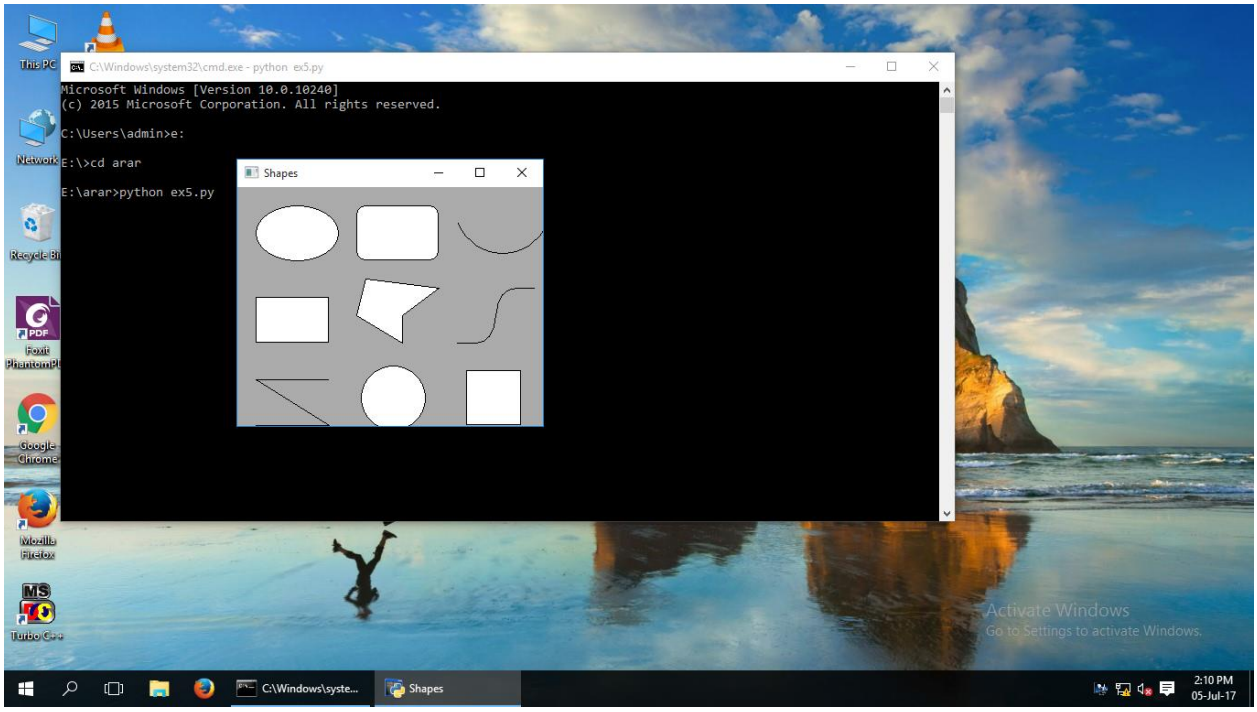
1. Start the program
2. Import the file “import wx”.
3. Create an “Class Shapes(wx.Frame)”.
4. Define the “def__init__(self,parent,id,title)”.
5. Design the “wxframe, Bind(wx.EVT_PAINT,self.Onpaint)”.
6. Define an “def.Onpaint(self,event)”.
7. Declare an shapes “Ellipse,RoundedRectangle,Polygon and Lines”.
8. Call an method “wx.app()” to print an Shapes.
9. Stop the program.

CODING:

```
import wx
class Shapes(wx.Frame):
def __init__(self, parent, id, title):
```

```
wx.Frame.__init__(self, parent, id, title, size=(350, 300))

self.Bind(wx.EVT_PAINT, self.OnPaint)

self.Centre()

self.Show(True)

def OnPaint(self, event):

dc = wx.PaintDC(self)

dc.DrawEllipse(20, 20, 90, 60)

dc.DrawRoundedRectangle(130, 20, 90, 60, 10)

dc.DrawArc(240, 40, 340, 40, 290, 20)

dc.DrawPolygon(((130, 140), (180, 170), (180, 140), (220, 110), (140, 100)))

dc.DrawRectangle(20, 120, 80, 50)

dc.DrawSpline(((240, 170), (280, 170), (285, 110), (325, 110)))

dc.DrawLines(((20, 260), (100, 260), (20, 210), (100, 210)))

dc.DrawCircle(170, 230, 35)

dc.DrawRectangle(250, 200, 60, 60)

app = wx.App()

Shapes(None, -1, 'Shapes')

app.MainLoop()
```

DRAWING AN API FOR GEOMETRIC SHAPE USING WXPYTHON



Ex. No.: 06 PERFORMING BASIC CLOUD STORAGE OPERATIONS USING PYTHON

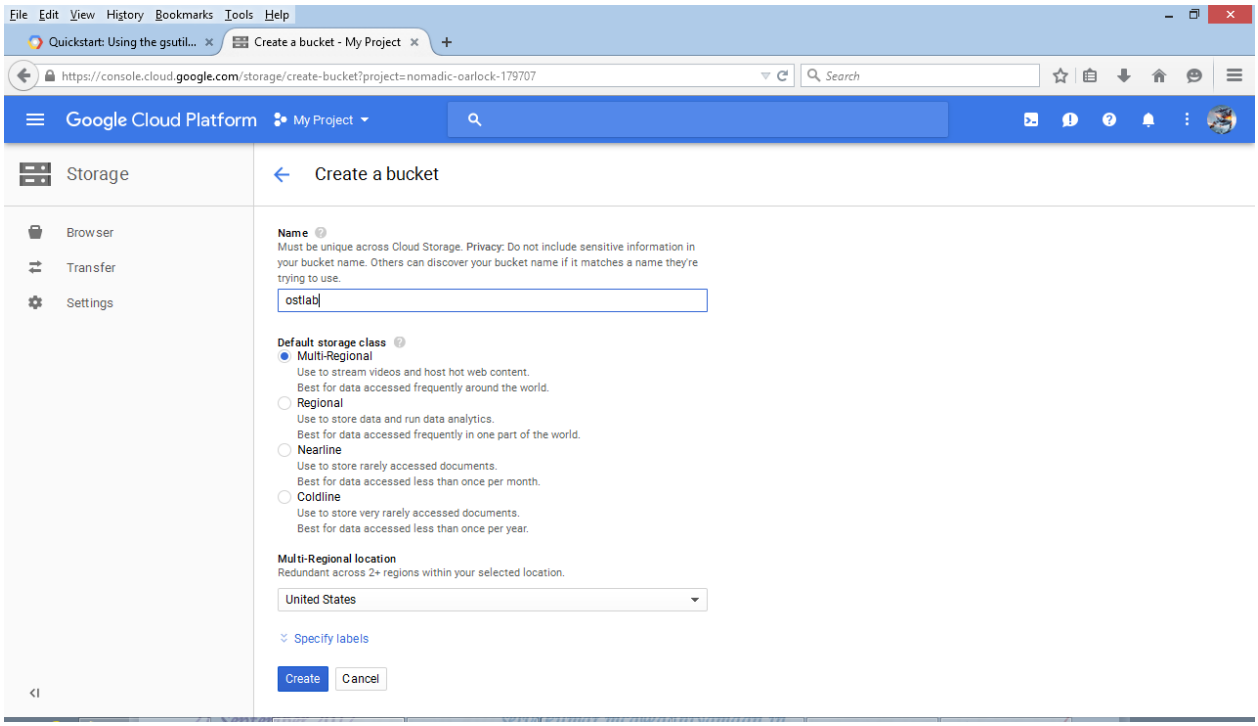
AIM:

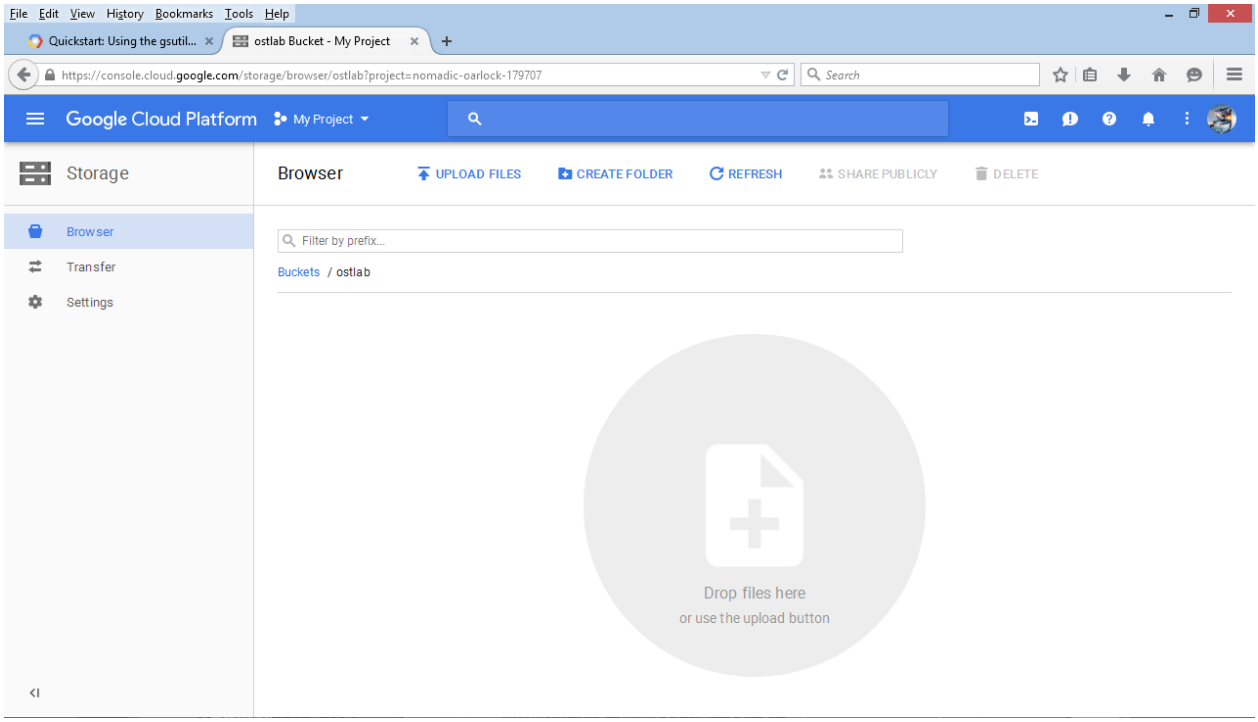
To perform a basic cloud storage operation using python gsutil.

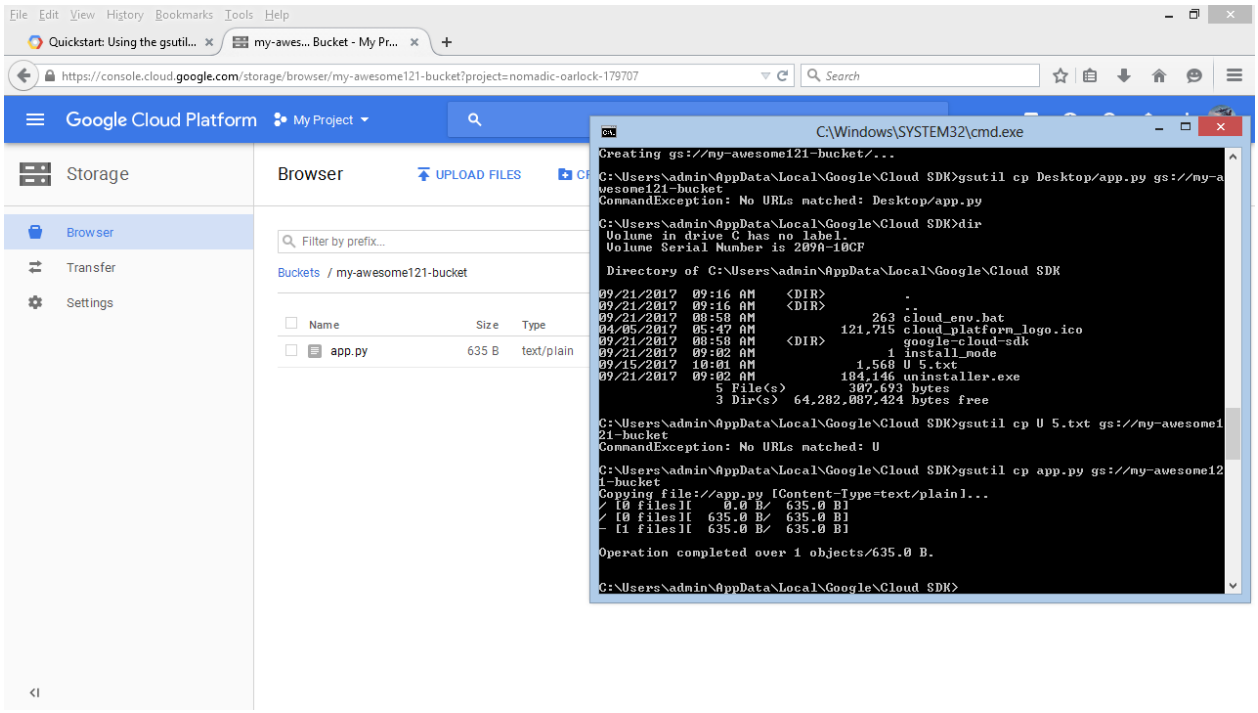
PROCEDURE:

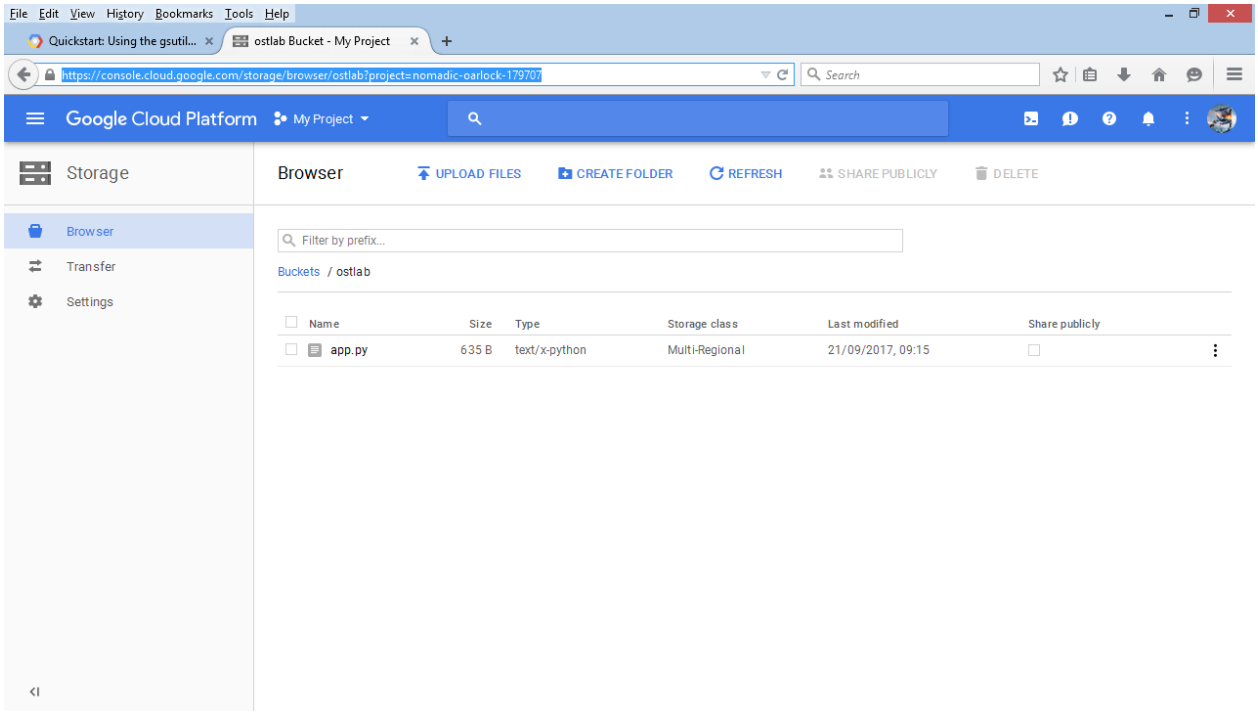
1. Start the program.
2. Create an account in Google cloud for cloud Storage.
3. Create a Bucket with necessary fields.
4. Bucket has been created and browse the file to be uploaded bucket of cloud storage.
5. Get a Cloud SDK by running the program on command prompt.
6. Stop the program.

PERFORMING BASIC CLOUD STORAGE OPERATION USING PYTHON GSUTIL






Quickstart: Using the gsutil...
my-awes... Bucket - My Pr...
https://console.cloud.google.com/storage/browser/my-awesome121-bucket?project=nomadic-oarlock-179707
Google Cloud Platform
My Project
Storage
Browser
Transfer
Settings
Browser
UPLOAD FILES
Filter by prefix...
Buckets / my-awesome121-bucket
Name
Size
Type
app.py
635 B
text/plain
C:\Windows\SYSTEM32\cmd.exe
Creating gs://my-awesome121-bucket/...
C:\Users\admin\AppData\Local\Google\Cloud SDK>gsutil cp Desktop/app.py gs://my-awesome121-bucket
CommandException: No URLs matched: Desktop/app.py
C:\Users\admin\AppData\Local\Google\Cloud SDK>dir
Volume in drive C has no label.
Volume Serial Number is 209A-10CF
Directory of C:\Users\admin\AppData\Local\Google\Cloud SDK
09/21/2017 09:16 AM <DIR> .
09/21/2017 09:16 AM <DIR> ..
09/21/2017 08:58 AM 263 cloud_env.bat
04/05/2017 05:47 AM 121,715 cloud_platform_logo.ico
09/21/2017 08:58 AM <DIR> google-cloud-sdk
09/21/2017 09:02 AM 1 install_mode
09/15/2017 10:01 AM 1,568 U 5.txt
09/21/2017 09:02 AM 184,146 uninstaller.exe
5 File(s) 307,693 bytes
3 Dir(s) 64,282,087,424 bytes free
C:\Users\admin\AppData\Local\Google\Cloud SDK>gsutil cp U 5.txt gs://my-awesome121-bucket
CommandException: No URLs matched: U
C:\Users\admin\AppData\Local\Google\Cloud SDK>gsutil cp app.py gs://my-awesome121-bucket
Copying file://app.py [Content-Type=text/plain]...
/ [0 files][0.0 B/ 635.0 B]
/ [0 files][635.0 B/ 635.0 B]
- [1 files][635.0 B/ 635.0 B]
Operation completed over 1 objects/635.0 B.
C:\Users\admin\AppData\Local\Google\Cloud SDK>


Quickstart: Using the gsutil...
ostlab Bucket - My Project
https://console.cloud.google.com/storage/browser/ostlab?project=nomadic-oarlock-179707
Google Cloud Platform
My Project
Storage
Browser
Transfer
Settings
Browser
UPLOAD FILES
CREATE FOLDER
REFRESH
SHARE PUBLICLY
DELETE
Filter by prefix...
Buckets / ostlab
Name
Size
Type
Storage class
Last modified
Share publicly
app.py
635 B
text/x-python
Multi-Regional
21/09/2017, 09:15

Ex. No.: 07 EXPLORE THE BASIC PLOT INTERFACE USING MATPLOTLIB

AIM:

To perform a basic plot interface using matplotlib.

PROCEDURE:

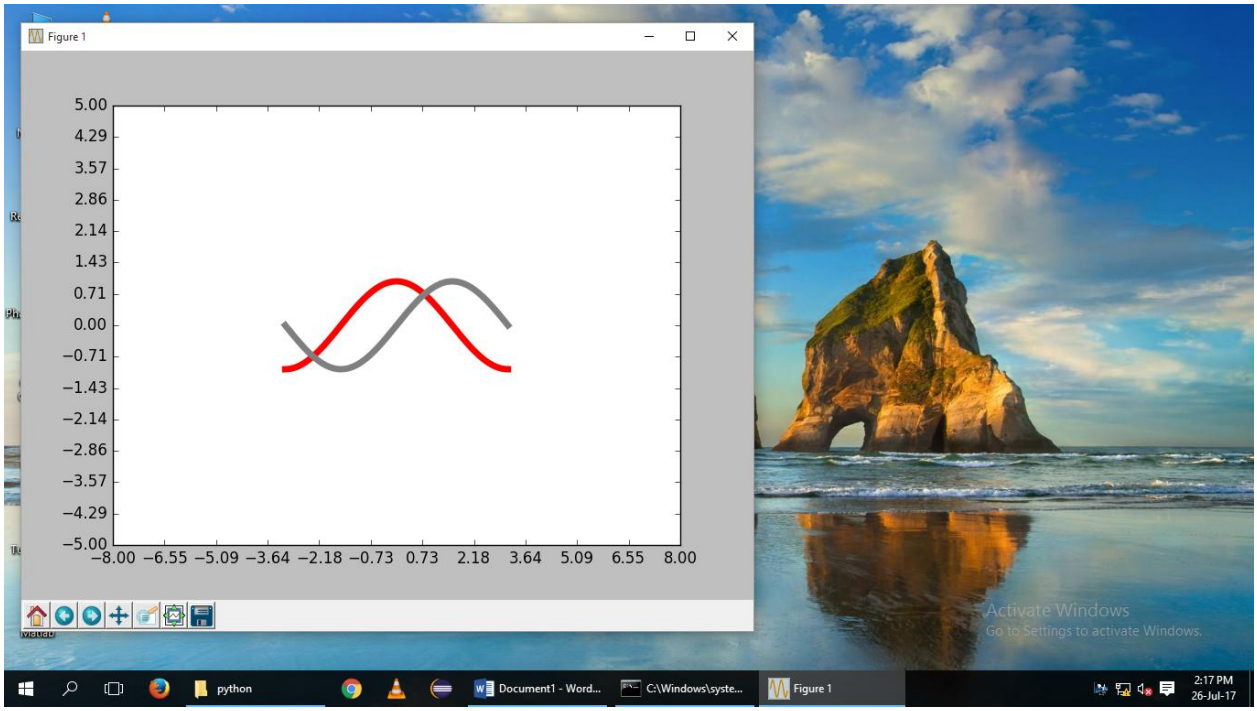
1. Start the program.
2. Import the file “import pandas as pd”.
3. Plot the figure using figuresize(7,5), dpi =70 & subplot(1,1,1).
4. Define the C,S as np.cos(x), np.sin(x).
5. Plot the cosine and sin with color, width and line style.
6. Plot the xlim & ylim with necessary values.
7. Create an Xtricks and ytricks (np ,linespace(4.2.0) and endpoint= true.
8. Call the method “plt.show()”.
9. Stop the program.

CODING:

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6), dpi=100)
```

```
plt.subplot(1, 1, 1)

X = np.linspace(-np.pi, np.pi, 256, endpoint=True)

C, S = np.cos(X), np.sin(X)

plt.plot(X, C, color="red", linewidth=5.0, linestyle="-")

plt.plot(X, S, color="gray", linewidth=5.0, linestyle="-")

plt.xlim(-4.0, 4.0)

plt.xticks(np.linspace(-8, 8, 12, endpoint=True))

plt.ylim(-1.0, 1.0)

plt.yticks(np.linspace(-5, 5, 15, endpoint=True))

plt.show()
```

EXPLORE THE BASIC PLOT INTERFACE USING MATPLOTLIB



Ex. No.: 08 PERFORMING DATA ANALYSIS USING PANDAS

AIM:

To perform data analysis using pandas.

PROCEDURE:

1. Start the program.
2. Import the file “import pandas as pd”.
3. Define “ipl_data(values to be printed)”.
4. Create an Dataframe(ipl_data).
5. Print the full dataframe values.
6. Use group, groupby statements to print particular values.
7. Stop the program.

CODING:

```
#import the pandas library
import pandas as pd
ipl_data = {'Team': ['Riders', 'Riders', 'Devils', 'Devils', 'Kings',
        'kings', 'Kings', 'Kings', 'Riders', 'Royals', 'Royals', 'Riders'],
        'Rank': [1, 2, 2, 3, 3,4 ,1 ,1,2 , 4,1,2],
```

```
        'Year': [2014,2015,2014,2015,2014,2015,2016,2017,2016,2014,2015,2017],
        'Points':[876,789,863,673,741,812,756,788,694,701,804,690]}
df = pd.DataFrame(ipl_data)          print df
ipl_data = {'Team': ['Riders', 'Riders', 'Devils', 'Devils', 'Kings',
        'kings', 'Kings', 'Kings', 'Riders', 'Royals', 'Royals', 'Riders'],
        'Rank': [1, 2, 2, 3, 3,4 ,1 ,1,2 , 4,1,2],
        'Year': [2014,2015,2014,2015,2014,2015,2016,2017,2016,2014,2015,2017],
        'Points':[876,789,863,673,741,812,756,788,694,701,804,690]}
df = pd.DataFrame(ipl_data)          grouped = df.groupby('Year')
for name,group in grouped:     print name      print group
ipl_data = {'Team': ['Riders', 'Riders', 'Devils', 'Devils', 'Kings',
        'kings', 'Kings', 'Kings', 'Riders', 'Royals', 'Royals', 'Riders'],
        'Rank': [1, 2, 2, 3, 3,4 ,1 ,1,2 , 4,1,2],
        'Year': [2014,2015,2014,2015,2014,2015,2016,2017,2016,2014,2015,2017],
        'Points':[876,789,863,673,741,812,756,788,694,701,804,690]}
df = pd.DataFrame(ipl_data)
grouped = df.groupby('Year')
print grouped.get_group(2014)
```

PERFORMING DATA ANALYSIS USING PANDAS