

Research Paper Easy Access

1.0

Generated by Doxygen 1.8.13

Contents

1	Main Page	1
2	Namespace Index	3
2.1	Packages	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	manage Namespace Reference	7
4.1.1	Detailed Description	7
4.1.2	Function Documentation	7
4.1.2.1	main()	7
4.2	settings Namespace Reference	8
4.2.1	Detailed Description	8
4.2.2	Variable Documentation	9
4.2.2.1	ALLOWED_HOSTS	9
4.2.2.2	AUTH_PASSWORD_VALIDATORS	9
4.2.2.3	BASE_DIR	9
4.2.2.4	CRISPY_TEMPLATE_PACK	9
4.2.2.5	DATABASES	10
4.2.2.6	DEBUG	10
4.2.2.7	INSTALLED_APPS	10
4.2.2.8	LANGUAGE_CODE	10
4.2.2.9	MEDIA_ROOT	11

4.2.2.10	MEDIA_URL	11
4.2.2.11	MIDDLEWARE	11
4.2.2.12	ROOT_URLCONF	11
4.2.2.13	SECRET_KEY	11
4.2.2.14	STATIC_ROOT	12
4.2.2.15	STATIC_URL	12
4.2.2.16	STATICFILES_DIRS	12
4.2.2.17	STATICFILES_FINDERS	12
4.2.2.18	TEMPLATES	12
4.2.2.19	TIME_ZONE	13
4.2.2.20	USE_I18N	13
4.2.2.21	USE_L10N	13
4.2.2.22	USE_TZ	13
4.2.2.23	WSGI_APPLICATION	13
4.3	urls Namespace Reference	13
4.3.1	Variable Documentation	13
4.3.1.1	urlpatterns	14
4.4	views Namespace Reference	14
4.4.1	Function Documentation	14
4.4.1.1	convert_pdf()	15
4.4.1.2	extractpdf_ajit()	16
4.4.1.3	extractpdf_ganesh()	16
4.4.1.4	extractpdf_mythili()	17
4.4.1.5	extractpdf_om()	17
4.4.1.6	extractpdf_pb()	18
4.4.1.7	extractpdf_prof()	18
4.4.1.8	extractpdf_rohit()	19
4.4.1.9	extractpdf_supratik()	19
4.4.1.10	extractpdf_varsha()	20
4.4.1.11	research_papers()	20
4.4.1.12	savepdf()	21
4.4.1.13	wc()	21
5	File Documentation	23
5.1	manage.py File Reference	23
5.2	settings.py File Reference	23
5.3	urls.py File Reference	24
5.4	views.py File Reference	24
Index		27

Chapter 1

Main Page

Documentation on How to run the code. To make the code run in your system, python and django should be installed to your system.

To install django:

```
$ sudo apt-get install django-python
```

OR

```
$ sudo apt install python3-pip
```

AND

```
$ pip3 install django
```

To install python:

```
$ sudo apt-get install python
```

Create a directory named 'papers'

start a project in terminal by using the following:

```
$ django -admin startproject research_project
```

A [manage.py](#) file will be created.

From paper folder to host the server:

```
$ python3 manage.py runserver
```

The code is in [views.py](#) and templates are being created in 'template' folder which is in research_paper.

Settings for the setting.py and [manage.py](#) file:

```
It has to be done as per specified in the setting.py file and as per requirement.
```

Server is setup. Now you can easily access!!

Author

Learners

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

manage	7
settings	8
urls	13
views	14

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

manage.py	23
settings.py	23
urls.py	24
views.py	24

Chapter 4

Namespace Documentation

4.1 manage Namespace Reference

Functions

- `def main ()`

Documentation of main function.

4.1.1 Detailed Description

Django's command-line utility for administrative tasks.

4.1.2 Function Documentation

4.1.2.1 `main()`

```
def manage.main ( )
```

Documentation of main function.

This function is used to set your [settings.py](#) file as default setting and the command is run from command line otherwise raise an error NO paramerts passed

4.2 settings Namespace Reference

Variables

- `BASE_DIR` = `os.path.dirname(os.path.dirname(os.path.abspath(__file__)))`
Documentation for the base directory path.
- string `SECRET_KEY` = `'-w-mk58zgjk*y=t61muzzrd+*e)7%+20j__!4_=lh(6^*@)42n'`
Documentation for security key.
- bool `DEBUG` = `True`
Documentation for security about DEBUG.
- list `ALLOWED_HOSTS` = `[]`
- list `INSTALLED_APPS`
To define Installed apps.
- string `CRISPY_TEMPLATE_PACK` = `'bootstrap4'`
- list `MIDDLEWARE`
To add the middleware required.
- string `ROOT_URLCONF` = `'papers.urls'`
Documentation for ROOT_URLconfig.
- list `TEMPLATES`
Documentation for Templates.
- string `WSGI_APPLICATION` = `'papers.wsgi.application'`
- dictionary `DATABASES`
- list `AUTH_PASSWORD_VALIDATORS`
Password validation.
- string `LANGUAGE_CODE` = `'en-us'`
for specifying the language code
- string `TIME_ZONE` = `'UTC'`
to specify the different parameters for setting of time zone for server
- bool `USE_I18N` = `True`
to specify the time zone for server
- bool `USE_L10N` = `True`
to specify the different parameters for setting of time zone for server
- bool `USE_TZ` = `True`
to specify the different parameters for setting of time zone for server
- tuple `STATICFILES_DIRS` = `(os.path.join(BASE_DIR, 'static'),)`
Documentation for static file directory path.
- tuple `STATICFILES_FINDERS`
Documentation for static file finders.
- string `STATIC_URL` = `'/static/'`
- `STATIC_ROOT` = `os.path.join(os.path.dirname(BASE_DIR), 'static_cdn')`
- `MEDIA_ROOT` = `os.path.join(BASE_DIR, 'media')`
- string `MEDIA_URL` = `'/media/'`

4.2.1 Detailed Description

Django settings for papers project.

Generated by 'django-admin startproject' using Django 2.2.6.

For more information on this file, see
<https://docs.djangoproject.com/en/2.2/topics/settings/>

For the full list of settings and their values, see
<https://docs.djangoproject.com/en/2.2/ref/settings/>

4.2.2 Variable Documentation

4.2.2.1 ALLOWED_HOSTS

```
list settings.ALLOWED_HOSTS = []
```

4.2.2.2 AUTH_PASSWORD_VALIDATORS

```
list settings.AUTH_PASSWORD_VALIDATORS
```

Initial value:

```
1 = [  
2     {  
3         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',  
4     },  
5     {  
6         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',  
7     },  
8     {  
9         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',  
10    },  
11    {  
12        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',  
13    },  
14 ]
```

Password validation.

It is used for password validation. Not required in Our case. <https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators>

4.2.2.3 BASE_DIR

```
settings.BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

Documentation for the base directory path.

It specifies the base path of the all files and folders of a project.

4.2.2.4 CRISPY_TEMPLATE_PACK

```
string settings.CRISPY_TEMPLATE_PACK = 'bootstrap4'
```

4.2.2.5 DATABASES

dictionary settings.DATABASES

Initial value:

```
1 = {
2     'default': {
3         'ENGINE': 'django.db.backends.sqlite3',
4         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
5     }
6 }
```

4.2.2.6 DEBUG

bool settings.DEBUG = True

Documentation for security about DEBUG.

to show the error status while development. Never use debug as true when used in production.

4.2.2.7 INSTALLED_APPS

list settings.INSTALLED_APPS

Initial value:

```
1 = [
2     'django.contrib.admin',
3     'django.contrib.auth',
4     'django.contrib.contenttypes',
5     'django.contrib.sessions',
6     'django.contrib.messages',
7     'django.contrib.staticfiles',
8     'research_papers',
9     'crispy_forms'
10 ]
```

To define Installed apps.

mentioning of all applications used, developer need to add his own project into it. In OUR case we added 'research paper'

4.2.2.8 LANGUAGE_CODE

string settings.LANGUAGE_CODE = 'en-us'

for specifying the language code

4.2.2.9 MEDIA_ROOT

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

4.2.2.10 MEDIA_URL

```
string settings.MEDIA_URL = '/media/'
```

4.2.2.11 MIDDLEWARE

```
list settings.MIDDLEWARE
```

Initial value:

```
1 = [  
2     'django.middleware.security.SecurityMiddleware',  
3     'django.contrib.sessions.middleware.SessionMiddleware',  
4     'django.middleware.common.CommonMiddleware',  
5     'django.middleware.csrf.CsrfViewMiddleware',  
6     'django.contrib.auth.middleware.AuthenticationMiddleware',  
7     'django.contrib.messages.middleware.MessageMiddleware',  
8     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
9 ]
```

To add the middleware required.

for our development No need of any change, it was already initialised.

4.2.2.12 ROOT_URLCONF

```
string settings.ROOT_URLCONF = 'papers.urls'
```

Documentation for ROOT_URLconfig.

to specify the path for urls file where the urlpatterns are being added.

4.2.2.13 SECRET_KEY

```
string settings.SECRET_KEY = '-w-mk58zgjk*y=t6lmuzzrd+*e)7%+20j__!4_=lh(6^*@)42n'
```

Documentation for security key.

Security key to be kept private when code used in production!!

4.2.2.14 STATIC_ROOT

```
settings.STATIC_ROOT = os.path.join( os.path.dirname( BASE_DIR ), 'static_cdn' )
```

4.2.2.15 STATIC_URL

```
string settings.STATIC_URL = '/static/'
```

4.2.2.16 STATICFILES_DIRS

```
tuple settings.STATICFILES_DIRS = (os.path.join( BASE_DIR, 'static' ),)
```

Documentation for static file directory path.

It is set to create a folder in base directory to store all the static files to be used such as images etc.

4.2.2.17 STATICFILES_FINDERS

```
tuple settings.STATICFILES_FINDERS
```

Initial value:

```
1 = (  
2 'django.contrib.staticfiles.finders.FileSystemFinder',  
3 'django.contrib.staticfiles.finders.AppDirectoriesFinder',)
```

Documentation for static file finders.

set to search for existence of any static files in project folder to be added

4.2.2.18 TEMPLATES

```
list settings.TEMPLATES
```

Initial value:

```
1 = [  
2     {  
3         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
4         'DIRS': [],  
5         'APP_DIRS': True,  
6         'OPTIONS': {  
7             'context_processors': [  
8                 'django.template.context_processors.debug',  
9                 'django.template.context_processors.request',  
10                'django.contrib.auth.context_processors.auth',  
11                'django.contrib.messages.context_processors.messages',  
12            ],  
13        },  
14    },  
15 ]
```

Documentation for Templates.

to specify the templates which has to be set to run the project.

4.2.2.19 TIME_ZONE

```
string settings.TIME_ZONE = 'UTC'
```

to specify the different parameters for setting of time zone for server

4.2.2.20 USE_I18N

```
bool settings.USE_I18N = True
```

to specify the time zone for server

4.2.2.21 USE_L10N

```
bool settings.USE_L10N = True
```

to specify the different parameters for setting of time zone for server

4.2.2.22 USE_TZ

```
bool settings.USE_TZ = True
```

to specify the different parameters for setting of time zone for server

4.2.2.23 WSGI_APPLICATION

```
string settings.WSGI_APPLICATION = 'papers.wsgi.application'
```

4.3 urls Namespace Reference

Variables

- list [urlpatterns](#)

Documentation for urlpatterns to specify the URL to be visited.

4.3.1 Variable Documentation

4.3.1.1 urlpatterns

```
list urls.urlpatterns
```

Initial value:

```
1 = [
2     path('', views.research_papers),
3     path('thankyou/', views.research_papers),
4
5 ]
```

Documentation for urlpatterns to specify the URL to be visited.

It is used to set the path for setting the pages or adding the pages in the project. it is setting the redirect of the first page to the page ("thankyou page) to result the output.

4.4 views Namespace Reference

Functions

- def [wc](#) ()
Documentation for [wc\(\)](#) function.
- def [convert_pdf](#) (path)
Documentation for [convertpdf](#) function.
- def [extractpdf_pb](#) (soup, add_url)
Documentation for [extractpdf_pb](#) function.
- def [extractpdf_varsha](#) (soup, add_url)
Documentation for [extractpdf_varsha](#) function.
- def [extractpdf_ganesh](#) (soup, add_url)
Documentation for [extractpdf_varsha](#) function.
- def [extractpdf_supratik](#) (soup, add_url)
Documentation for [extractpdf_supratik](#) function.
- def [extractpdf_mythili](#) (soup, add_url)
Documentation for [extractpdf_mythili](#) function.
- def [extractpdf_rohit](#) (soup, add_url)
Documentation for [extractpdf_rohit](#) function.
- def [extractpdf_om](#) (soup, add_url)
Documentation for [extractpdf_om](#) function.
- def [extractpdf_ajit](#) (soup, add_url)
Documentation for [extractpdf_ajit](#) function.
- def [extractpdf_prof](#) (soup, URL)
Documentation for [extractpdf_ajit](#) function.
- def [savepdf](#) (list_pdf)
Documentation for [savepdf](#) function.
- def [research_papers](#) (request)
Documentation for [research_paper](#) function.

4.4.1 Function Documentation

4.4.1.1 `convert_pdf()`

```
def views.convert_pdf (  
    path )
```

Documentation for convertpdf function.

Parameters

<i>path</i>	It takes the path of file which has to converted to string . the input file is of pdf type. This function opens the pdf file inside it and convert the file into string
-------------	---

Returns

str which is a string storing the whole file content.

4.4.1.2 extractpdf_ajit()

```
def views.extractpdf_ajit (
    soup,
    add_url )
```

Documentation for extractpdf_ajit function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. Ajit Diwan).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.3 extractpdf_ganesh()

```
def views.extractpdf_ganesh (
    soup,
    add_url )
```

Documentation for extractpdf_varsha function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof Ganesh Ramakrishnan).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.4 extractpdf_mythili()

```
def views.extractpdf_mythili (
    soup,
    add_url )
```

Documentation for extractpdf_mythili function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. Mythili Vutukuru).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.5 extractpdf_om()

```
def views.extractpdf_om (
    soup,
    add_url )
```

Documentation for extractpdf_om function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. Om Damani).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.6 extractpdf_pb()

```
def views.extractpdf_pb (
    soup,
    add_url )
```

Documentation for extractpdf_pb function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. Pushpak Bhattacharya).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.7 extractpdf_prof()

```
def views.extractpdf_prof (
    soup,
    URL )
```

Documentation for extractpdf_ajit function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>URL</i>	It is the base url of the Professors page This function figures out the function to be called on the basis of given input URL.

This function calls the `extractpdf` function as per professor URL which returns the list of pdf mapped to URLs. If any URL for which no function mapping is yet done is passed, it print error message.

Returns

`url_pdf` it is the list of dictionary containing the paper name along with its URL

4.4.1.8 `extractpdf_rohit()`

```
def views.extractpdf_rohit (
    soup,
    add_url )
```

Documentation for `extractpdf_rohit` function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. Rohit Gurjar).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

`url_pdf` it is the list of dictionary containing the paper name along with its URL.

4.4.1.9 `extractpdf_supratik()`

```
def views.extractpdf_supratik (
    soup,
    add_url )
```

Documentation for `extractpdf_supratik` function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. supratik Bhattacharya).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch

the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.10 extractpdf_varsha()

```
def views.extractpdf_varsha (
    soup,
    add_url )
```

Documentation for extractpdf_varsha function.

Parameters

<i>soup</i>	It contains the html page fetched by the request function which is converted to html5lib format using beautiful soup
<i>add_url</i>	It is the base url of the Professors page This function process the content of html page to extract the name and links of the pdf of research paper publish by the faculty(Prof. Varsha Apte).

This function visit to the actual page where papers list has been displayed with the help of beautifulsoup and fetch the paper name along with the url to download the paper. The details for a single paper is stored in a dictionary with key as 'name' and 'url' and form a list of such dictionaries

Returns

url_pdf it is the list of dictionary containing the paper name along with its URL.

4.4.1.11 research_papers()

```
def views.research_papers (
    request )
```

Documentation for research_paper function.

Parameters

<i>request</i>	this function is passed with request which is used to get the requested posted on page This function process the URL and return the processed words to be displayed inthe form of word cloud.
----------------	---

On recieving the URL from User it calls the functions to extract the list of pdf links and download them using savpdf function. Afterwards, It will call convertpdf to convert pdf to string and then store it as text file to make word count and the whole procedure is repeated for all the files. then the top 50 occuring words are choosen to be displayed.

Returns

response It return the response to the page after processing the request in form of word cloud.

4.4.1.12 savepdf()

```
def views.savepdf (
    list_pdf )
```

Documentation for savepdf function.

Parameters

<i>list_pdf</i>	It contains the list of dictionary storing the pdf name mapped to its URL. This function will store all the pdf into system for further processing.
-----------------	---

It creates a directory named pdfstore where all the pdf are being stored as pdf0.pdf, pdf1.pdf and so on.

Returns

It return nothing. display a message saved when all pdf has been downloaded.

4.4.1.13 wc()

```
def views.wc ( )
```

Documentation for [wc\(\)](#) function.

This function opens a csv file inside it which contains a list of word along with its frequencies.

It then saves a image of word cloud generated at specified location with different word size. This function is not used by us in current development but if image word cloud is needed, function code can be uncommented.

```
reader = csv.reader(open('counts.csv', 'r'))
d = {}
for k,v in reader:
    #insert the frequency values in d array
    d[k] = float(v)

#creates a dummy mask(shape) for wordcloud
#mask = np.array(Image.open("images.jpeg"))
#word cloud corresponding to the word whose frequencies are in d array will be generated
wordcloud = WordCloud(width=5000, height=6000,max_words=400,stopwords=STOPWORDS,
random_state=42).generate_from_frequencies(d)
wordcloud.to_file('papers/static/image/images.png') #saves the wordcloud to image.png file
```


Chapter 5

File Documentation

5.1 manage.py File Reference

Namespaces

- [manage](#)

Functions

- `def manage.main ()`
Documentation of main function.

5.2 settings.py File Reference

Namespaces

- [settings](#)

Variables

- `settings.BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))`
Documentation for the base directory path.
- `string settings.SECRET_KEY = '-w-mk58zjgk*y=t61muzzrd+*e)7%+20j__!4_=lh(6^*@)42n'`
Documentation for security key.
- `bool settings.DEBUG = True`
Documentation for security about DEBUG.
- `list settings.ALLOWED_HOSTS = []`
- `list settings.INSTALLED_APPS`
To define Installed apps.
- `string settings.CRISPY_TEMPLATE_PACK = 'bootstrap4'`
- `list settings.MIDDLEWARE`
To add the middleware required.
- `string settings.ROOT_URLCONF = 'papers.urls'`

Documentation for ROOT_URLconfig.

- list [settings.TEMPLATES](#)

Documentation for Templates.

- string [settings.WSGI_APPLICATION](#) = 'papers.wsgi.application'
- dictionary [settings.DATABASES](#)
- list [settings.AUTH_PASSWORD_VALIDATORS](#)

Password validation.

- string [settings.LANGUAGE_CODE](#) = 'en-us'

for specifying the language code

- string [settings.TIME_ZONE](#) = 'UTC'

to specify the different parameters for setting of time zone for server

- bool [settings.USE_I18N](#) = True

to specify the time zone for server

- bool [settings.USE_L10N](#) = True

to specify the different parameters for setting of time zone for server

- bool [settings.USE_TZ](#) = True

to specify the different parameters for setting of time zone for server

- tuple [settings.STATICFILES_DIRS](#) = (os.path.join(BASE_DIR, 'static'),)

Documentation for static file directory path.

- tuple [settings.STATICFILES_FINDERS](#)

Documentation for static file finders.

- string [settings.STATIC_URL](#) = '/static/'
- [settings.STATIC_ROOT](#) = os.path.join(os.path.dirname(BASE_DIR), 'static_cdn')
- [settings.MEDIA_ROOT](#) = os.path.join(BASE_DIR, 'media')
- string [settings.MEDIA_URL](#) = '/media/'

5.3 urls.py File Reference

Namespaces

- [urls](#)

Variables

- list [urls.urlpatterns](#)

Documentation for urlpatterns to specify the URL to be visited.

5.4 views.py File Reference

Namespaces

- [views](#)

Functions

- def [views.wc](#) ()
Documentation for [wc\(\)](#) function.
- def [views.convert_pdf](#) (path)
Documentation for [convertpdf](#) function.
- def [views.extractpdf_pb](#) (soup, add_url)
Documentation for [extractpdf_pb](#) function.
- def [views.extractpdf_varsha](#) (soup, add_url)
Documentation for [extractpdf_varsha](#) function.
- def [views.extractpdf_ganesh](#) (soup, add_url)
Documentation for [extractpdf_varsha](#) function.
- def [views.extractpdf_supratik](#) (soup, add_url)
Documentation for [extractpdf_supratik](#) function.
- def [views.extractpdf_mythili](#) (soup, add_url)
Documentation for [extractpdf_mythili](#) function.
- def [views.extractpdf_rohit](#) (soup, add_url)
Documentation for [extractpdf_rohit](#) function.
- def [views.extractpdf_om](#) (soup, add_url)
Documentation for [extractpdf_om](#) function.
- def [views.extractpdf_ajit](#) (soup, add_url)
Documentation for [extractpdf_ajit](#) function.
- def [views.extractpdf_prof](#) (soup, URL)
Documentation for [extractpdf_ajit](#) function.
- def [views.savepdf](#) (list_pdf)
Documentation for [savepdf](#) function.
- def [views.research_papers](#) (request)
Documentation for [research_paper](#) function.

Index

ALLOWED_HOSTS

settings, [9](#)

AUTH_PASSWORD_VALIDATORS

settings, [9](#)

BASE_DIR

settings, [9](#)

CRISPY_TEMPLATE_PACK

settings, [9](#)

convert_pdf

views, [14](#)

DATABASES

settings, [9](#)

DEBUG

settings, [10](#)

extractpdf_ajit

views, [16](#)

extractpdf_ganesh

views, [16](#)

extractpdf_mythili

views, [17](#)

extractpdf_om

views, [17](#)

extractpdf_pb

views, [18](#)

extractpdf_prof

views, [18](#)

extractpdf_rohit

views, [19](#)

extractpdf_supratik

views, [19](#)

extractpdf_varsha

views, [20](#)

INSTALLED_APPS

settings, [10](#)

LANGUAGE_CODE

settings, [10](#)

MEDIA_ROOT

settings, [10](#)

MEDIA_URL

settings, [11](#)

MIDDLEWARE

settings, [11](#)

main

manage, [7](#)

manage, [7](#)

main, [7](#)

manage.py, [23](#)

ROOT_URLCONF

settings, [11](#)

research_papers

views, [20](#)

SECRET_KEY

settings, [11](#)

STATIC_ROOT

settings, [11](#)

STATIC_URL

settings, [12](#)

STATICFILES_DIRS

settings, [12](#)

STATICFILES_FINDERS

settings, [12](#)

savepdf

views, [21](#)

settings, [8](#)

ALLOWED_HOSTS, [9](#)

AUTH_PASSWORD_VALIDATORS, [9](#)

BASE_DIR, [9](#)

CRISPY_TEMPLATE_PACK, [9](#)

DATABASES, [9](#)

DEBUG, [10](#)

INSTALLED_APPS, [10](#)

LANGUAGE_CODE, [10](#)

MEDIA_ROOT, [10](#)

MEDIA_URL, [11](#)

MIDDLEWARE, [11](#)

ROOT_URLCONF, [11](#)

SECRET_KEY, [11](#)

STATIC_ROOT, [11](#)

STATIC_URL, [12](#)

STATICFILES_DIRS, [12](#)

STATICFILES_FINDERS, [12](#)

TEMPLATES, [12](#)

TIME_ZONE, [12](#)

USE_I18N, [13](#)

USE_L10N, [13](#)

USE_TZ, [13](#)

WSGI_APPLICATION, [13](#)

settings.py, [23](#)

TEMPLATES

settings, [12](#)

TIME_ZONE

- settings, [12](#)
- USE_I18N
 - settings, [13](#)
- USE_L10N
 - settings, [13](#)
- USE_TZ
 - settings, [13](#)
- urlpatterns
 - urls, [13](#)
- urls, [13](#)
 - urlpatterns, [13](#)
- urls.py, [24](#)
- views, [14](#)
 - convert_pdf, [14](#)
 - extractpdf_ajit, [16](#)
 - extractpdf_ganesh, [16](#)
 - extractpdf_mythili, [17](#)
 - extractpdf_om, [17](#)
 - extractpdf_pb, [18](#)
 - extractpdf_prof, [18](#)
 - extractpdf_rohit, [19](#)
 - extractpdf_supratik, [19](#)
 - extractpdf_varsha, [20](#)
 - research_papers, [20](#)
 - savepdf, [21](#)
 - wc, [21](#)
- views.py, [24](#)
- WSGI_APPLICATION
 - settings, [13](#)
- wc
 - views, [21](#)