

settings.py

Django settings for question_answer_system project.

Generated by 'django-admin startproject' using Django 2.2.3.

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/>

Build paths inside the project like this: `os.path.join(BASE_DIR, ...)`

Quick-start development settings - unsuitable for production See

<https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/>

SECURITY WARNING: keep the secret key used in production secret!

SECURITY WARNING: don't run with debug turned on in production!

Application definition

django middleware definitions

```
import os
```

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

```
SECRET_KEY = 'cd93pyyv6)ax+s%t)8noto5$+l3&w$*eu-e8#u%&-zowy$^yn! '
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'whoosh',  
    'haystack',  
    'django_extensions',  
    'qsapp',  
]
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',
```

root url definition

templates/html files path definitions along with templating engine

wsgi , needed when deploying the website with apache/nginx

Database <https://docs.djangoproject.com/en/2.2/ref/settings/#databases>

whoosh index directory definition

haystack realtime processor means rebuild index as soon as some records are added

haystack engine definition for whoosh

```
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'question_answer_system.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

```
WSGI_APPLICATION = 'question_answer_system.wsgi.application'
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

```
WHOOSH_INDEX = os.path.join(BASE_DIR, 'whoosh/')
```

```
HAYSTACK_SIGNAL_PROCESSOR = 'haystack.signals.RealtimeSignalProcessor'
```

```
HAYSTACK_CONNECTIONS = {
    'default': {
        'ENGINE': 'haystack.backends.whoosh_backend.WhooshEngine',
        'PATH': WHOOSH_INDEX,
```

Password validation <https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators>

Internationalization <https://docs.djangoproject.com/en/2.2/topics/i18n/>

Static files (CSS, JavaScript, Images)

<https://docs.djangoproject.com/en/2.2/howto/static-files/>

will not be served, more like long term storage

long term storage for the static files

```
    },
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static-storage')
]

STATIC_ROOT = os.path.join(BASE_DIR, 'static-serve')
```