

ASSIGNMENT ON SEARCH BASED SOFTWARE ENGINEERING AND COMBINATORIAL INTERACTION TESTING

JUSTYNA PETKE

Answers should be send by e-mail to j.petke@ucl.ac.uk by **January 15, 2016**.
Please typeset your manuscript in LaTeX.

1. Software development effort estimation is the process of predicting the most realistic amount of effort (expressed in terms of person-hours or money or software size) required to develop or maintain software based on incomplete, uncertain and noisy input. Given the following problem:

Given a set of well-formed equations which can be used to produce cost predictions and a set of real-world data, find the equation with best predictive capability.

Answer the following questions:

- (1) What are the objectives?
- (2) How an example solution would look like, i.e., what is the problem representation?
- (3) How would you measure the fitness function for a single objective?
- (4) Which search algorithm would you choose? Why?

2. Describe two other software engineering problems to which SBSE techniques can be applied to. Describe why the problems are fit for SBSE (hint: think about the search space). Describe the problem representation (including how would a neighbouring solution look like) and propose a fitness function.

3. Generate a 4-way interaction test suite for the following set of configurations for a Car Model:

Automated Driving Controller	Collision Avoidance Braking	Parallel Parking	Lateral Range Finder
Included None	StandardAvoidance EnhancedAvoidance None	Included None	Included None

Constraints:

- Parallel parking must always be included.
- If automated driving controller is included, then collision avoidance braking must be enhanced.
- If collision avoidance braking is enhanced, then lateral range finder must be included.

4. Django is a very popular Python-based framework for rapid development of web-based applications. Among popular sites using it are: Instagram, Mozilla and The Washington Times. Django is written in Python and comes with a global settings file that contains parameters that can be configured in any Django-based web application. Given the settings file at https://github.com/django/django/blob/master/django/conf/global_settings.py (also presented in Appendix A below), answer the following questions:

- (1) How many parameters are there in the Django settings file?
- (2) List all the Boolean parameters.
- (3) Create a minimal pairwise test suite for the Boolean parameters.

APPENDIX A. DJANGO'S GLOBAL_SETTINGS.PY FILE.

```

# Default Django settings. Override these with settings in the module
# pointed-to by the DJANGO_SETTINGS_MODULE environment variable.

# This is defined here as a do-nothing function because we can't import
# django.utils.translation -- that module depends on the settings.
gettext_noop = lambda s: s

#####
# CORE #
#####

DEBUG = False
TEMPLATE_DEBUG = False

# Whether the framework should propagate raw exceptions rather than catching
# them. This is useful under some testing situations and should never be used
# on a live site.
DEBUG_PROPAGATE_EXCEPTIONS = False

# Whether to use the "Etag" header. This saves bandwidth but slows down performance.
USE_ETAGS = False

# People who get code error notifications.
# In the format [('Full Name', 'email@example.com'), ('Full Name', 'anotheremail@example.com')]
ADMINS = []

# List of IP addresses, as strings, that:
# * See debug comments, when DEBUG is true
# * Receive x-headers
INTERNAL_IPS = []

# Hosts/domain names that are valid for this site.
# "*" matches anything, ".example.com" matches example.com and all subdomains
ALLOWED_HOSTS = []

# Local time zone for this installation. All choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name (although not all
# systems may support all possibilities). When USE_TZ is True, this is
# interpreted as the default user time zone.
TIME_ZONE = 'America/Chicago'

# If you set this to True, Django will use timezone-aware datetimes.
USE_TZ = False

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = 'en-us'

# Languages we provide translations for, out of the box.
LANGUAGES = [
    ('af', gettext_noop('Afrikaans')),
    ('ar', gettext_noop('Arabic')),
    ('ast', gettext_noop('Asturian')),
    ('az', gettext_noop('Azerbaijani')),
    ('bg', gettext_noop('Bulgarian')),
    ('be', gettext_noop('Belarusian')),
    ('bn', gettext_noop('Bengali')),
    ('br', gettext_noop('Breton')),
    ('bs', gettext_noop('Bosnian')),
    ('ca', gettext_noop('Catalan')),
    ('cs', gettext_noop('Czech')),
    ('cy', gettext_noop('Welsh')),
    ('da', gettext_noop('Danish')),
    ('de', gettext_noop('German')),
    ('el', gettext_noop('Greek')),
    ('en', gettext_noop('English')),

```

```

('en-au', gettext_noop('Australian English')),
('en-gb', gettext_noop('British English')),
('eo', gettext_noop('Esperanto')),
('es', gettext_noop('Spanish')),
('es-ar', gettext_noop('Argentinian Spanish')),
('es-mx', gettext_noop('Mexican Spanish')),
('es-ni', gettext_noop('Nicaraguan Spanish')),
('es-ve', gettext_noop('Venezuelan Spanish')),
('et', gettext_noop('Estonian')),
('eu', gettext_noop('Basque')),
('fa', gettext_noop('Persian')),
('fi', gettext_noop('Finnish')),
('fr', gettext_noop('French')),
('fy', gettext_noop('Frisian')),
('ga', gettext_noop('Irish')),
('gl', gettext_noop('Galician')),
('he', gettext_noop('Hebrew')),
('hi', gettext_noop('Hindi')),
('hr', gettext_noop('Croatian')),
('hu', gettext_noop('Hungarian')),
('ia', gettext_noop('Interlingua')),
('id', gettext_noop('Indonesian')),
('io', gettext_noop('Ido')),
('is', gettext_noop('Icelandic')),
('it', gettext_noop('Italian')),
('ja', gettext_noop('Japanese')),
('ka', gettext_noop('Georgian')),
('kk', gettext_noop('Kazakh')),
('km', gettext_noop('Khmer')),
('kn', gettext_noop('Kannada')),
('ko', gettext_noop('Korean')),
('lb', gettext_noop('Luxembourgish')),
('lt', gettext_noop('Lithuanian')),
('lv', gettext_noop('Latvian')),
('mk', gettext_noop('Macedonian')),
('ml', gettext_noop('Malayalam')),
('mn', gettext_noop('Mongolian')),
('mr', gettext_noop('Marathi')),
('my', gettext_noop('Burmese')),
('nb', gettext_noop('Norwegian Bokmal')),
('ne', gettext_noop('Nepali')),
('nl', gettext_noop('Dutch')),
('nn', gettext_noop('Norwegian Nynorsk')),
('os', gettext_noop('Ossetic')),
('pa', gettext_noop('Punjabi')),
('pl', gettext_noop('Polish')),
('pt', gettext_noop('Portuguese')),
('pt-br', gettext_noop('Brazilian Portuguese')),
('ro', gettext_noop('Romanian')),
('ru', gettext_noop('Russian')),
('sk', gettext_noop('Slovak')),
('sl', gettext_noop('Slovenian')),
('sq', gettext_noop('Albanian')),
('sr', gettext_noop('Serbian')),
('sr-latn', gettext_noop('Serbian Latin')),
('sv', gettext_noop('Swedish')),
('sw', gettext_noop('Swahili')),
('ta', gettext_noop('Tamil')),
('te', gettext_noop('Telugu')),
('th', gettext_noop('Thai')),
('tr', gettext_noop('Turkish')),
('tt', gettext_noop('Tatar')),
('udm', gettext_noop('Udmurt')),
('uk', gettext_noop('Ukrainian')),
('ur', gettext_noop('Urdu')),
('vi', gettext_noop('Vietnamese')),
('zh-hans', gettext_noop('Simplified Chinese')),
('zh-hant', gettext_noop('Traditional Chinese')),
]

```

ASSIGNMENT ON SEARCH BASED SOFTWARE ENGINEERING AND COMBINATORIAL INTERACTION TESTING

```
# Languages using BiDi (right-to-left) layout
LANGUAGES_BIDI = ["he", "ar", "fa", "ur"]

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True
LOCALE_PATHS = []

# Settings for language cookie
LANGUAGE_COOKIE_NAME = 'django_language'
LANGUAGE_COOKIE_AGE = None
LANGUAGE_COOKIE_DOMAIN = None
LANGUAGE_COOKIE_PATH = '/'

# If you set this to True, Django will format dates, numbers and calendars
# according to user current locale.
USE_L10N = False

# Not-necessarily-technical managers of the site. They get broken link
# notifications and other various emails.
MANAGERS = ADMINS

# Default content type and charset to use for all HttpResponse objects, if a
# MIME type isn't manually specified. These are used to construct the
# Content-Type header.
DEFAULT_CONTENT_TYPE = 'text/html'
DEFAULT_CHARSET = 'utf-8'

# Encoding of files read from disk (template and initial SQL files).
FILE_CHARSET = 'utf-8'

# Email address that error messages come from.
SERVER_EMAIL = 'root@localhost'

# Database connection info. If left empty, will default to the dummy backend.
DATABASES = {}

# Classes used to implement DB routing behavior.
DATABASE_ROUTERS = []

# The email backend to use. For possible shortcuts see django.core.mail.
# The default is to use the SMTP backend.
# Third-party backends can be specified by providing a Python path
# to a module that defines an EmailBackend class.
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

# Host for sending email.
EMAIL_HOST = 'localhost'

# Port for sending email.
EMAIL_PORT = 25

# Optional SMTP authentication information for EMAIL_HOST.
EMAIL_HOST_USER = ''
EMAIL_HOST_PASSWORD = ''
EMAIL_USE_TLS = False
EMAIL_USE_SSL = False
EMAIL_SSL_CERTFILE = None
EMAIL_SSL_KEYFILE = None
EMAIL_TIMEOUT = None

# List of strings representing installed apps.
INSTALLED_APPS = []

# List of locations of the template source files, in search order.
TEMPLATE_DIRS = []
```

```

# List of callables that know how to import templates from various sources.
# See the comments in django/core/template/loader.py for interface
# documentation.
TEMPLATE_LOADERS = [
    'django.template.loaders.filesystem.Loader',
    'django.template.loaders.app_directories.Loader',
    # 'django.template.loaders.eggs.Loader',
]

# List of processors used by RequestContext to populate the context.
# Each one should be a callable that takes the request object as its
# only parameter and returns a dictionary to add to the context.
TEMPLATE_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.template.context_processors.request',
    'django.contrib.messages.context_processors.messages',
]

# Output to use in template system for invalid (e.g. misspelled) variables.
TEMPLATE_STRING_IF_INVALID = ''

TEMPLATES = []

# Default email address to use for various automated correspondence from
# the site managers.
DEFAULT_FROM_EMAIL = 'webmaster@localhost'

# Subject-line prefix for email messages send with django.core.mail.mail_admins
# or ...mail_managers. Make sure to include the trailing space.
EMAIL_SUBJECT_PREFIX = '[Django] '

# Whether to append trailing slashes to URLs.
APPEND_SLASH = True

# Whether to prepend the "www." subdomain to URLs that don't have it.
PREPEND_WWW = False

# Override the server-derived value of SCRIPT_NAME
FORCE_SCRIPT_NAME = None

# List of compiled regular expression objects representing User-Agent strings
# that are not allowed to visit any page, systemwide. Use this for bad
# robots/crawlers. Here are a few examples:
# import re
# DISALLOWED_USER_AGENTS = [
#     re.compile(r'^NaverBot.*'),
#     re.compile(r'^EmailSiphon.*'),
#     re.compile(r'^SiteSucker.*'),
#     re.compile(r'^sohu-search')
# ]
DISALLOWED_USER_AGENTS = []

ABSOLUTE_URL_OVERRIDES = {}

# List of strings representing allowed prefixes for the {% ssi %} tag.
# Example: ['/home/html', '/var/www']
ALLOWED_INCLUDE_ROOTS = []

# List of compiled regular expression objects representing URLs that need not
# be reported by BrokenLinkEmailsMiddleware. Here are a few examples:
# import re
# IGNOREABLE_404_URLS = [
#     re.compile(r'^/apple-touch-icon.*\.png$'),
#     re.compile(r'^/favicon.ico$'),

```

ASSIGNMENT ON SEARCH BASED SOFTWARE ENGINEERING AND COMBINATORIAL INTERACTION TESTING

```

#         re.compile(r'^/robots.txt$),
#         re.compile(r'^/phpmyadmin/),
#         re.compile(r'\.(cgi|php|pl)$'),
#     ]
IGNORABLE_404_URLS = []

# A secret key for this particular Django installation. Used in secret-key
# hashing algorithms. Set this in your settings, or Django will complain
# loudly.
SECRET_KEY = ''

# Default file storage mechanism that holds media.
DEFAULT_FILE_STORAGE = 'django.core.files.storage.FileSystemStorage'

# Absolute filesystem path to the directory that will hold user-uploaded files.
# Example: "/var/www/example.com/media/"
MEDIA_ROOT = ''

# URL that handles the media served from MEDIA_ROOT.
# Examples: "http://example.com/media/", "http://media.example.com/"
MEDIA_URL = ''

# Absolute path to the directory static files should be collected to.
# Example: "/var/www/example.com/static/"
STATIC_ROOT = None

# URL that handles the static files served from STATIC_ROOT.
# Example: "http://example.com/static/", "http://static.example.com/"
STATIC_URL = None

# List of upload handler classes to be applied in order.
FILE_UPLOAD_HANDLERS = [
    'django.core.files.uploadhandler.MemoryFileUploadHandler',
    'django.core.files.uploadhandler.TemporaryFileUploadHandler',
]

# Maximum size, in bytes, of a request before it will be streamed to the
# file system instead of into memory.
FILE_UPLOAD_MAX_MEMORY_SIZE = 2621440 # i.e. 2.5 MB

# Directory in which upload streamed files will be temporarily saved. A value of
# 'None' will make Django use the operating system's default temporary directory
# (i.e. "/tmp" on *nix systems).
FILE_UPLOAD_TEMP_DIR = None

# The numeric mode to set newly-uploaded files to. The value should be a mode
# you'd pass directly to os.chmod; see http://docs.python.org/lib/os-file-dir.html.
FILE_UPLOAD_PERMISSIONS = None

# The numeric mode to assign to newly-created directories, when uploading files.
# The value should be a mode as you'd pass to os.chmod;
# see http://docs.python.org/lib/os-file-dir.html.
FILE_UPLOAD_DIRECTORY_PERMISSIONS = None

# Python module path where user will place custom format definition.
# The directory where this setting is pointing should contain subdirectories
# named as the locales, containing a formats.py file
# (i.e. "myproject.locale" for myproject/locale/en/formats.py etc. use)
FORMAT_MODULE_PATH = None

# Default formatting for date objects. See all available format strings here:
# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
DATE_FORMAT = 'N j, Y'

# Default formatting for datetime objects. See all available format strings here:
# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
DATETIME_FORMAT = 'N j, Y, P'

# Default formatting for time objects. See all available format strings here:

```

```

# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
TIME_FORMAT = 'P'

# Default formatting for date objects when only the year and month are relevant.
# See all available format strings here:
# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
YEAR_MONTH_FORMAT = 'F Y'

# Default formatting for date objects when only the month and day are relevant.
# See all available format strings here:
# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
MONTH_DAY_FORMAT = 'F j'

# Default short formatting for date objects. See all available format strings here:
# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
SHORT_DATE_FORMAT = 'm/d/Y'

# Default short formatting for datetime objects.
# See all available format strings here:
# http://docs.djangoproject.com/en/dev/ref/templates/builtins/#date
SHORT_DATETIME_FORMAT = 'm/d/Y P'

# Default formats to be used when parsing dates from input boxes, in order
# See all available format string here:
# http://docs.python.org/library/datetime.html#strptime-behavior
# * Note that these format strings are different from the ones to display dates
DATE_INPUT_FORMATS = [
    '%Y-%m-%d', '%m/%d/%Y', '%m/%d/%y', # '2006-10-25', '10/25/2006', '10/25/06'
    '%b %d %Y', '%b %d, %Y', # 'Oct 25 2006', 'Oct 25, 2006'
    '%d %b %Y', '%d %b, %Y', # '25 Oct 2006', '25 Oct, 2006'
    '%B %d %Y', '%B %d, %Y', # 'October 25 2006', 'October 25, 2006'
    '%B %d %Y', '%d %B, %Y', # '25 October 2006', '25 October, 2006'
]

# Default formats to be used when parsing times from input boxes, in order
# See all available format string here:
# http://docs.python.org/library/datetime.html#strptime-behavior
# * Note that these format strings are different from the ones to display dates
TIME_INPUT_FORMATS = [
    '%H:%M:%S', # '14:30:59'
    '%H:%M:%S.%f', # '14:30:59.000200'
    '%H:%M', # '14:30'
]

# Default formats to be used when parsing dates and times from input boxes,
# in order
# See all available format string here:
# http://docs.python.org/library/datetime.html#strptime-behavior
# * Note that these format strings are different from the ones to display dates
DATETIME_INPUT_FORMATS = [
    '%Y-%m-%d %H:%M:%S', # '2006-10-25 14:30:59'
    '%Y-%m-%d %H:%M:%S.%f', # '2006-10-25 14:30:59.000200'
    '%Y-%m-%d %H:%M', # '2006-10-25 14:30'
    '%Y-%m-%d', # '2006-10-25'
    '%m/%d/%Y %H:%M:%S', # '10/25/2006 14:30:59'
    '%m/%d/%Y %H:%M:%S.%f', # '10/25/2006 14:30:59.000200'
    '%m/%d/%Y %H:%M', # '10/25/2006 14:30'
    '%m/%d/%Y', # '10/25/2006'
    '%m/%d/%y %H:%M:%S', # '10/25/06 14:30:59'
    '%m/%d/%y %H:%M:%S.%f', # '10/25/06 14:30:59.000200'
    '%m/%d/%y %H:%M', # '10/25/06 14:30'
    '%m/%d/%y', # '10/25/06'
]

# First day of week, to be used on calendars
# 0 means Sunday, 1 means Monday...
FIRST_DAY_OF_WEEK = 0

# Decimal separator symbol

```


ASSIGNMENT ON SEARCH BASED SOFTWARE ENGINEERING AND COMBINATORIAL INTERACTION TESTING 9

```
DECIMAL_SEPARATOR = '.'

# Boolean that sets whether to add thousand separator when formatting numbers
USE_THOUSAND_SEPARATOR = False

# Number of digits that will be together, when splitting them by
# THOUSAND_SEPARATOR. 0 means no grouping, 3 means splitting by thousands...
NUMBER_GROUPING = 0

# Thousand separator symbol
THOUSAND_SEPARATOR = ','

# The tablespaces to use for each model when not specified otherwise.
DEFAULT_TABLESPACE = ''
DEFAULT_INDEX_TABLESPACE = ''

# Default X-Frame-Options header value
X_FRAME_OPTIONS = 'SAMEORIGIN'

USE_X_FORWARDED_HOST = False

# The Python dotted path to the WSGI application that Django's internal server
# (runserver) will use. If 'None', the return value of
# 'django.core.wsgi.get_wsgi_application' is used, thus preserving the same
# behavior as previous versions of Django. Otherwise this should point to an
# actual WSGI application object.
WSGI_APPLICATION = None

# If your Django app is behind a proxy that sets a header to specify secure
# connections, AND that proxy ensures that user-submitted headers with the
# same name are ignored (so that people can't spoof it), set this value to
# a tuple of (header_name, header_value). For any requests that come in with
# that header/value, request.is_secure() will return True.
# WARNING! Only set this if you fully understand what you're doing. Otherwise,
# you may be opening yourself up to a security risk.
SECURE_PROXY_SSL_HEADER = None

#####
# MIDDLEWARE #
#####

# List of middleware classes to use. Order is important; in the request phase,
# this middleware classes will be applied in the order given, and in the
# response phase the middleware will be applied in reverse order.
MIDDLEWARE_CLASSES = [
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
]

#####
# SESSIONS #
#####

# Cache to store session data if using the cache session backend.
SESSION_CACHE_ALIAS = 'default'
# Cookie name. This can be whatever you want.
SESSION_COOKIE_NAME = 'sessionid'
# Age of cookie, in seconds (default: 2 weeks).
SESSION_COOKIE_AGE = 60 * 60 * 24 * 7 * 2
# A string like ".example.com", or None for standard domain cookie.
SESSION_COOKIE_DOMAIN = None
# Whether the session cookie should be secure (https:// only).
SESSION_COOKIE_SECURE = False
# The path of the session cookie.
SESSION_COOKIE_PATH = '/'
# Whether to use the non-RFC standard httpOnly flag (IE, FF3+, others)
SESSION_COOKIE_HTTPONLY = True
# Whether to save the session data on every request.
SESSION_SAVE_EVERY_REQUEST = False
```

```

# Whether a user's session cookie expires when the Web browser is closed.
SESSION_EXPIRE_AT_BROWSER_CLOSE = False
# The module to store session data
SESSION_ENGINE = 'django.contrib.sessions.backends.db'
# Directory to store session files if using the file session module. If None,
# the backend will use a sensible default.
SESSION_FILE_PATH = None
# class to serialize session data
SESSION_SERIALIZER = 'django.contrib.sessions.serializers.JSONSerializer'

#####
# CACHE #
#####

# The cache backends to use.
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.locmem.LocMemCache',
    }
}
CACHE_MIDDLEWARE_KEY_PREFIX = ''
CACHE_MIDDLEWARE_SECONDS = 600
CACHE_MIDDLEWARE_ALIAS = 'default'

#####
# AUTHENTICATION #
#####

AUTH_USER_MODEL = 'auth.User'

AUTHENTICATION_BACKENDS = ['django.contrib.auth.backends.ModelBackend']

LOGIN_URL = '/accounts/login/'

LOGOUT_URL = '/accounts/logout/'

LOGIN_REDIRECT_URL = '/accounts/profile/'

# The number of days a password reset link is valid for
PASSWORD_RESET_TIMEOUT_DAYS = 3

# the first hasher in this list is the preferred algorithm. any
# password using different algorithms will be converted automatically
# upon login
PASSWORD_HASHERS = [
    'django.contrib.auth.hashers.PBKDF2PasswordHasher',
    'django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher',
    'django.contrib.auth.hashers.BCryptSHA256PasswordHasher',
    'django.contrib.auth.hashers.BCryptPasswordHasher',
    'django.contrib.auth.hashers.SHA1PasswordHasher',
    'django.contrib.auth.hashers.MD5PasswordHasher',
    'django.contrib.auth.hashers.UnsaltedSHA1PasswordHasher',
    'django.contrib.auth.hashers.UnsaltedMD5PasswordHasher',
    'django.contrib.auth.hashers.CryptPasswordHasher',
]

#####
# SIGNING #
#####

SIGNING_BACKEND = 'django.core.signing.TimestampSigner'

#####
# CSRF #
#####

# Dotted path to callable to be used as view when a request is
# rejected by the CSRF middleware.
CSRF_FAILURE_VIEW = 'django.views.csrf.csrf_failure'

```

ASSIGNMENT ON SEARCH BASED SOFTWARE ENGINEERING AND COMBINATORIAL INTERACTION TESTING

```
# Settings for CSRF cookie.
CSRF_COOKIE_NAME = 'csrftoken'
CSRF_COOKIE_AGE = 60 * 60 * 24 * 7 * 52
CSRF_COOKIE_DOMAIN = None
CSRF_COOKIE_PATH = '/'
CSRF_COOKIE_SECURE = False
CSRF_COOKIE_HTTPONLY = False
CSRF_HEADER_NAME = 'HTTP_X_CSRFTOKEN'

#####
# MESSAGES #
#####

# Class to use as messages backend
MESSAGE_STORAGE = 'django.contrib.messages.storage.fallback.FallbackStorage'

# Default values of MESSAGE_LEVEL and MESSAGE_TAGS are defined within
# django.contrib.messages to avoid imports in this settings file.

#####
# LOGGING #
#####

# The callable to use to configure logging
LOGGING_CONFIG = 'logging.config.dictConfig'

# Custom logging configuration.
LOGGING = {}

# Default exception reporter filter class used in case none has been
# specifically assigned to the HttpRequest instance.
DEFAULT_EXCEPTION_REPORTER_FILTER = 'django.views.debug.SafeExceptionReporterFilter'

#####
# TESTING #
#####

# The name of the class to use to run the test suite
TEST_RUNNER = 'django.test.runner.DiscoverRunner'

# Apps that don't need to be serialized at test database creation time
# (only apps with migrations are to start with)
TEST_NON_SERIALIZED_APPS = []

#####
# FIXTURES #
#####

# The list of directories to search for fixtures
FIXTURE_DIRS = []

#####
# STATICFILES #
#####

# A list of locations of additional static files
STATICFILES_DIRS = []

# The default file storage backend used during the build process
STATICFILES_STORAGE = 'django.contrib.staticfiles.storage.StaticFilesStorage'

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = [
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
    # 'django.contrib.staticfiles.finders.DefaultStorageFinder',
]
]
```

```
#####  
# MIGRATIONS #  
#####  
  
# Migration module overrides for apps, by app label.  
MIGRATION_MODULES = {}  
  
#####  
# SYSTEM CHECKS #  
#####  
  
# List of all issues generated by system checks that should be silenced. Light  
# issues like warnings, infos or debugs will not generate a message. Silencing  
# serious issues like errors and criticals does not result in hiding the  
# message, but Django will not stop you from e.g. running server.  
SILENCED_SYSTEM_CHECKS = []  
  
#####  
# SECURITY MIDDLEWARE #  
#####  
SECURE_BROWSER_XSS_FILTER = False  
SECURE_CONTENT_TYPE_NOSNIFF = False  
SECURE_HSTS_INCLUDE_SUBDOMAINS = False  
SECURE_HSTS_SECONDS = 0  
SECURE_REDIRECT_EXEMPT = []  
SECURE_SSL_HOST = None  
SECURE_SSL_REDIRECT = False
```