Contents

1 MySQL connection

1. install mysqlclient
2. set mysql as database engine
3. defaults -> OPTION -> ”read_default_file”: ”path/to/mysql.cnf”

2 Admin Panel

3 Models

3.1 verbose_name

3.1.1 Each field type, except for Keys

```
first_name = models.CharField("person's first name", max_length=30)
```

3.1.2 for Keys field

```
sites = models.ManyToManyField(Site, verbose_name="list of sites")
```

we can use this approach for the other field type too.

3.1.3 for Model

create subclass Meta in the model then add verbose_name and verbose_name_plural attributes.

3.2 on_delete

3.2.1 models.CASCADE

deletes the object containing the ForeignKey.(when parent delete related children delete too)

3.2.2 models.SET_NULL

Set the ForeignKey null; this is only possible if null is True.(if parent delete children set the foreign key null)

3.2.3 models.Protected

Prevent the deletion, first delete children then delete parent.

3.3 Relationships

best practice = name the foreign key field be the name of the model, lowercase

3.3.1 One To Many

```
one_to_many_field = models.ForeignKey(ParentClass, on_delete)
```

recursive relationships to create recursive relationships; an object that has a one to many relationship with it self do like this

```
models.ForeignKey('self', on_delete=models.CASCADE)
```

4 Managers

models.models.field() here models is a manager in models create class for self defined manages which inherit from models.Manage e.g, manager for published articles

5 Installing an App

add 'app_name' or 'app_name.apps.MainConfig' to the INSTALLEDAPP in the setting with the second approach we add some configuration to the app like verbose_name

6 Farsi language and Tehran Time

LANGUAGE_CODE = 'fa-ir' TIME_ZONE = 'Asia/Tehran'

7 Views

7.1 generic DetailView

Attributes	
model	model that want to create form for it
fields	a list off model field to represent in the form
template_name	form template name

7.1.1 How to send data(e.g, pictures) throw django using HTML forms

set encrypt to multipart/form-data in the form html element.

7.2 model.get_absolute_url

absolute url for the model.

7.2.1 use it in the templates

{{instance.get_absolute_url}}

8 Authentication

admin and auth app in installed apps
field error and non field error

8.1 Create a Login Page

- create an app
- use django contrib.auth urls
- create a template for login page(default django view looks for registration/login.html)
- create view and template for login_redirect_urls
- login_redirect_url in settings.py
- login required decorators or mixins

8.1.1 LOGIN_URL

a variable in the django settings file, where the request are redirect when using login_required decorators or LoginRequired mixin

8.1.2 LOGIN_REDIRECT_URL

a variable in the django settings file, where request are redirect after login if LoginView doesn't get a next Get parameter.

8.2 login_required decorator

redirect request to the settings.LOGIN_URL if user isn't logged in.

8.3 LoginRequired Mixin

can achieve the same behavior as with login_required by using LoginRequired mixin.

9 Templates

9.1 Static Files

9.1.1 STATICFILES_DIRS

its a additional location that static_files app will traverse if the FileSystemFinder is enabled, e.g. if you use the collectstatic or findstatic management command or use the static file serving view
it's default values is an empty list
STATICFILES_DIRS = [
BASE_DIR / ”static”,
’/var/www/static/’,
]

9.1.2 include tag

use to load and renders it in the current context. templates name []. pass a value to a template form another template [].

9.1.3 load static and static tag

{%load static%} use whenever call a static file

href = ” {% static ’url’ %} ”

9.2 Django Tweak

tweak the form field rendering in templates no in python level form definitions, altering CSS classes and html attributes are supported.

9.3 Fields Error

9.4 Non Fields Error

like wrong password or username

9.4.1 Disable JavaScript if Firefox

use about:config

9.5 include tag

9.6 block tag

9.7 url tag

{% url 'main:student_course_list' user.get_username %} for parameter don't use quote

9.8 truncate tag tezmplate filter

9.9 foreign key value in a template

10 ORM

11 and, or in query

12 foreign key value in the query

13 Form

13.1 Raw Html Form

13.2 Pure Django Form

13.3 Django Model Form

13.4 form validation