

python django

By Rishi



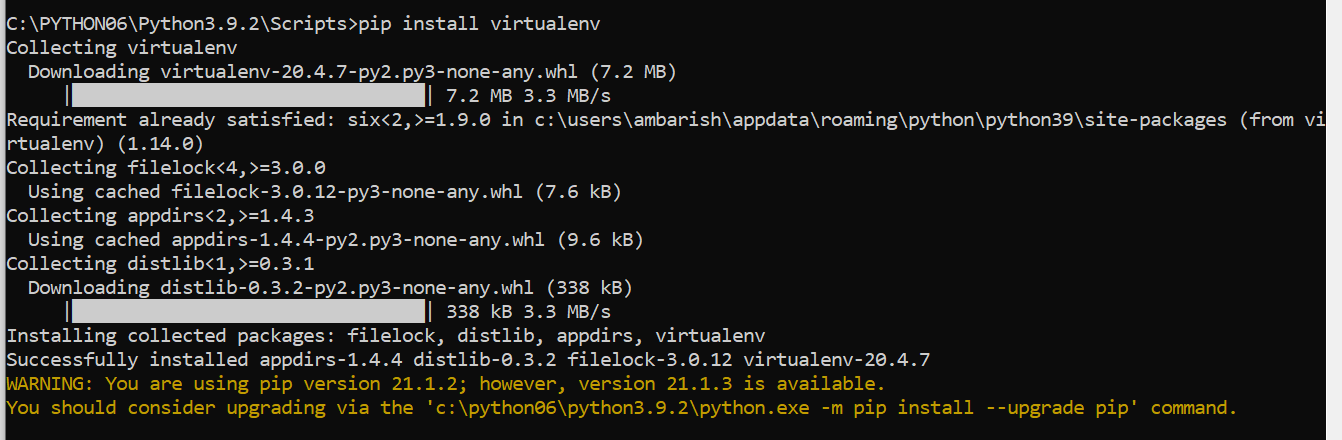
December 14, 2021

rishi.h2kinfosys@gmail.com

www.h2kinfosys.com

*Installing Virtual Environment:*

*pip install virtualenv*

**

*C:\>cd Rishi*

*C:\RISHI>cd Dev*

*C:\RISHI\Dev>mkdir myDjang*

*C:\RISHI\Dev>dir*

*Volume in drive C is Windows*

*Volume Serial Number is 0CDC-A763*

*Directory of C:\RISHI\Dev*

*31-07-2019 21:38 <DIR> .*

*31-07-2019 21:38 <DIR> ..*

*31-07-2019 21:38 <DIR> myDjang*

*0 File(s) 0 bytes*

*8 Dir(s) 575,761,174,528 bytes free*

*C:\RISHI\Dev>cd myDjang*

*C:\RISHI\Dev\aprDjango>virtualenv .*

*created virtual environment CPython3.8.2.final.0-32 in 19295ms*

*creator CPython3Windows(dest=C:\RISHI\Dev\aprDjango, clear=False, global=False)*

*seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, via=copy, app\_data\_dir=C:\Users\Rishi\AppData\Local\pypa\virtualenv\seed-app-data\v1.0.1)*

*activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator*

*C:\RISHI\Dev\myDjang>dir*

*Volume in drive C is Windows*

*Volume Serial Number is 0CDC-A763*

*Directory of C:\RISHI\Dev\myDjang*

*31-07-2019 21:39 <DIR> .*

*31-07-2019 21:39 <DIR> ..*

*16-04-2019 11:20 <DIR> Include*

*31-07-2019 21:39 <DIR> Lib*

*28-03-2018 17:07 30,340 LICENSE.txt*

*31-07-2019 21:40 <DIR> Scripts*

*31-07-2019 21:39 <DIR> tcl*

*1 File(s) 30,340 bytes*

*6 Dir(s) 575,724,306,432 bytes free*

*C:\RISHI\Dev\myDjang>cd Scripts*

*C:\RISHI\Dev\myDjang\Scripts>dir*

*Volume in drive C is Windows*

*Volume Serial Number is 0CDC-A763*

*Directory of C:\RISHI\Dev\myDjang\Scripts*

*31-07-2019 21:40 <DIR> .*

*31-07-2019 21:40 <DIR> ..*

*31-07-2019 21:40 2,315 activate*

*31-07-2019 21:40 883 activate.bat*

*31-07-2019 21:40 2,038 activate.ps1*

*31-07-2019 21:40 1,159 activate.xsh*

*31-07-2019 21:40 1,517 activate\_this.py*

*31-07-2019 21:40 512 deactivate.bat*

*31-07-2019 21:40 102,783 easy\_install-3.6.exe*

*31-07-2019 21:40 102,783 easy\_install.exe*

*31-07-2019 21:40 102,765 pip.exe*

*31-07-2019 21:40 102,765 pip3.6.exe*

*31-07-2019 21:40 102,765 pip3.exe*

*31-07-2019 21:39 100,504 python.exe*

*31-07-2019 21:39 58,520 python3.dll*

*31-07-2019 21:39 3,610,776 python36.dll*

*31-07-2019 21:39 98,968 pythonw.exe*

*31-07-2019 21:40 102,761 wheel.exe*

*16 File(s) 4,493,814 bytes*

*2 Dir(s) 575,724,306,432 bytes free*

*C:\RISHI\Dev\myDjang\Scripts>activate*

*(myDjang) C:\RISHI\Dev\myDjang\Scripts>*

*(myDjang) C:\RISHI\Dev\myDjang\Scripts>cd ..*

*(myDjang) C:\RISHI\Dev\myDjang>mkdir src*

*(myDjang) C:\RISHI\Dev\myDjang>dir*

*Volume in drive C is Windows*

*Volume Serial Number is 0CDC-A763*

*Directory of C:\RISHI\Dev\myDjang*

*31-07-2019 21:43 <DIR> .*

*31-07-2019 21:43 <DIR> ..*

*16-04-2019 11:20 <DIR> Include*

*31-07-2019 21:39 <DIR> Lib*

*28-03-2018 17:07 30,340 LICENSE.txt*

*31-07-2019 21:40 <DIR> Scripts*

*31-07-2019 21:43 <DIR> src*

*31-07-2019 21:39 <DIR> tcl*

*1 File(s) 30,340 bytes*

*7 Dir(s) 575,725,363,200 bytes free*

*(aprDjango) C:\RISHI\Dev\aprDjango\src>pip install django*

*Collecting django*

*Downloading Django-3.0.6-py3-none-any.whl (7.5 MB)*

*|████████████████████████████████| 7.5 MB 3.3 MB/s*

*Collecting asgiref~=3.2*

*Using cached asgiref-3.2.7-py2.py3-none-any.whl (19 kB)*

*Collecting sqlparse>=0.2.2*

*Using cached sqlparse-0.3.1-py2.py3-none-any.whl (40 kB)*

*Collecting pytz*

*Using cached pytz-2020.1-py2.py3-none-any.whl (510 kB)*

*Installing collected packages: asgiref, sqlparse, pytz, django*

*Successfully installed asgiref-3.2.7 django-3.0.6 pytz-2020.1 sqlparse-0.3.1*

*WARNING: You are using pip version 20.0.2; however, version 20.1 is available.*

*You should consider upgrading via the 'C:\RISHI\Dev\aprDjango\Scripts\python.exe -m pip install --upgrade pip' command.*

*(myDjang) C:\RISHI\Dev\myDjang\ src >*

*(myDjang) C:\RISHI\Dev\myDjang\src >pip freeze*

*Django==2.0.7*

*pytz==2019.2*

*(myDjang) C:\RISHI\Dev\myDjang>cd src*

*(myDjang) C:\RISHI\Dev\myDjang\src>django-admin*

*Type 'django-admin help <subcommand>' for help on a specific subcommand.*

*Available subcommands:*

*[django]*

*check*

*compilemessages*

*createcachetable*

*dbshell*

*diffsettings*

*dumpdata*

*flush*

*inspectdb*

*loaddata*

*makemessages*

*makemigrations*

*migrate*

*runserver*

*sendtestemail*

*shell*

*showmigrations*

*sqlflush*

*sqlmigrate*

*sqlsequencereset*

*squashmigrations*

*startapp*

*startproject*

*test*

*testserver*

Note that only Django core commands are listed as settings are not properly configured (error: Requested setting INSTALLED\_APPS, but settings are not configured. You must either define the environment variable DJANGO\_SETTINGS\_MODULE or call settings.configure() before accessing settings.).

(*myDjang) C:\RISHI\Dev\myDjang\src>django-admin startproject myDjang .*

*(myDjang) C:\RISHI\Dev\myDjang\src>dir*

*Volume in drive C is Windows*

*Volume Serial Number is 0CDC-A763*

*Directory of C:\RISHI\Dev\myDjang\src*

*31-07-2019 21:50 <DIR> .*

*31-07-2019 21:50 <DIR> ..*

*31-07-2019 21:50 <DIR> myDjang*

*31-07-2019 21:50 559 manage.py*

*1 File(s) 559 bytes*

*3 Dir(s) 575,693,553,664 bytes free*

*(aprDjango) C:\RISHI\Dev\aprDjango\src>python manage.py runserver*

*Watching for file changes with StatReloader*

*Performing system checks...*

*System check identified no issues (0 silenced).*

*You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.*

*Run 'python manage.py migrate' to apply them.*

*May 04, 2020 - 20:42:08*

*Django version 3.0.6, using settings 'aprDjango.settings'*

*Starting development server at http://127.0.0.1:8000/*

*Quit the server with CTRL-BREAK.*

*Open second command prompt and execute below command:*

*(septDjango) C:\RISHI\Dev\septDjango\src>python manage.py migrate*

*Operations to perform:*

*Apply all migrations: admin, auth, contenttypes, sessions*

*Running migrations:*

*Applying contenttypes.0001\_initial... OK*

*Applying auth.0001\_initial... OK*

*Applying admin.0001\_initial... OK*

*Applying admin.0002\_logentry\_remove\_auto\_add... OK*

*Applying contenttypes.0002\_remove\_content\_type\_name... OK*

*Applying auth.0002\_alter\_permission\_name\_max\_length... OK*

*Applying auth.0003\_alter\_user\_email\_max\_length... OK*

*Applying auth.0004\_alter\_user\_username\_opts... OK*

*Applying auth.0005\_alter\_user\_last\_login\_null... OK*

*Applying auth.0006\_require\_contenttypes\_0002... OK*

*Applying auth.0007\_alter\_validators\_add\_error\_messages... OK*

*Applying auth.0008\_alter\_user\_username\_max\_length... OK*

*Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK*

*Applying sessions.0001\_initial... OK*

*(septDjango) C:\RISHI\Dev\septDjango\src>python manage.py makemigrations*

*No changes detected*

*(myDjang) C:\RISHI\Dev\myDjang\src>python manage.py createsuperuser*

*Username: admin*

*Email address: admin@myDjang.com*

*Password:*

*Password (again):*

*Superuser created successfully.*

*(myDjang) C:\RISHI\Dev\myDjang\src>python manage.py runserver*

*Performing system checks...*

System check identified no issues (0 silenced).

July 31, 2019 - 22:07:54

Django version 2.0.7, using settings 'myDjang.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

[31/Jul/2019 22:08:08] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0

[31/Jul/2019 22:08:09] "GET /admin/ HTTP/1.1" 200 2984

[31/Jul/2019 22:08:09] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 412

[31/Jul/2019 22:08:09] "GET /static/admin/css/responsive.css HTTP/1.1" 304 0

[31/Jul/2019 22:08:09] "GET /static/admin/css/base.css HTTP/1.1" 304 0

[31/Jul/2019 22:08:09] "GET /static/admin/css/fonts.css HTTP/1.1" 304 0

[31/Jul/2019 22:08:09] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0

[31/Jul/2019 22:08:09] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0

[31/Jul/2019 22:08:09] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0

[31/Jul/2019 22:08:09] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331

[31/Jul/2019 22:08:09] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380

[31/Jul/2019 22:08:21] "GET /admin/auth/user/ HTTP/1.1" 200 7089

[31/Jul/2019 22:08:21] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6170

[31/Jul/2019 22:08:21] "GET /admin/jsi18n/ HTTP/1.1" 200 3185

[31/Jul/2019 22:08:22] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 363

[31/Jul/2019 22:08:22] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 258648

[31/Jul/2019 22:08:22] "GET /static/admin/js/core.js HTTP/1.1" 200 7134

[31/Jul/2019 22:08:22] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 6897

[31/Jul/2019 22:08:22] "GET /static/admin/js/actions.js HTTP/1.1" 200 6502

[31/Jul/2019 22:08:22] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1538

[31/Jul/2019 22:08:22] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 128820

[31/Jul/2019 22:08:22] "GET /static/admin/js/urlify.js HTTP/1.1" 200 8729

[31/Jul/2019 22:08:22] "GET /static/admin/img/search.svg HTTP/1.1" 200 458

[31/Jul/2019 22:08:22] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436

[31/Jul/2019 22:08:22] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331

[31/Jul/2019 22:08:22] "GET /static/admin/img/sorting-icons.svg HTTP/1.1" 200 1097

[31/Jul/2019 22:08:40] "GET /admin/ HTTP/1.1" 200 2984

[31/Jul/2019 22:08:42] "GET /admin/auth/group/ HTTP/1.1" 200 3584

[31/Jul/2019 22:08:42] "GET /admin/jsi18n/ HTTP/1.1" 200 3185

[31/Jul/2019 22:08:46] "GET /admin/ HTTP/1.1" 200 2984

Performing system checks...

System check identified no issues (0 silenced).

July 31, 2019 - 22:17:44

Django version 2.0.7, using settings 'myDjang.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

## On Another Command Prompt

(*myDjang) C:\RISHI\Dev\myDjang\src>python manage.py makemigrations*

*(myDjang) C:\RISHI\Dev\myDjang\src>python manage.py migrate*

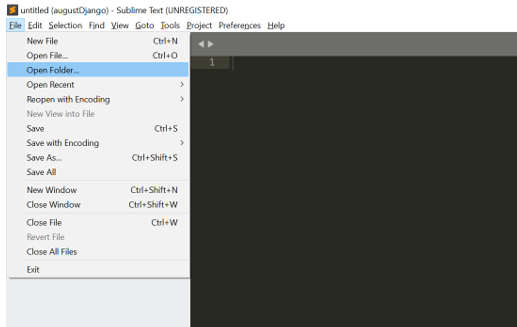
*(myDjang) C:\RISHI\Dev\myDjang\src>*

<https://docs.djangoproject.com/en/2.2/ref/models/fields/>

Important note for you, download Subline text: <https://www.sublimetext.com/3>

Django Project Creation with Eclipse:

1. Download and install Eclipse: <https://www.eclipse.org/downloads/packages/>
2. Download “Eclipse for Enterprise Java Developers”
3. Once installed, start eclipse and Install PyDev : <https://www.youtube.com/watch?v=r8joszYxPqk>



Use this option to navigate to Virtual environment folder and open the project here.

# Understanding Settings:

BASE\_DIR – this is where your project is.

DEBUG = True 🡪 Very important for development. Don’t deploy this in production

INSTALLED\_APPS 🡪 think of apps more like a component you are going to develop

ROOT\_URLCONF 🡪 Context Root of application

TEMPLATES 🡪 Configure your HTML pages here. We will learn more along the way

DATABASES 🡪 Django by default maps sql lite 3. You can map your database instance here

AUTH\_PASSWORD\_VALIDATORS 🡪 Password authenticators

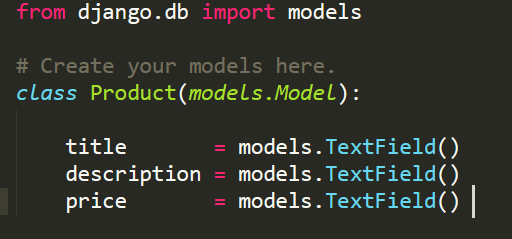
# Creating your Own App:

Creating your own applications (Apps):

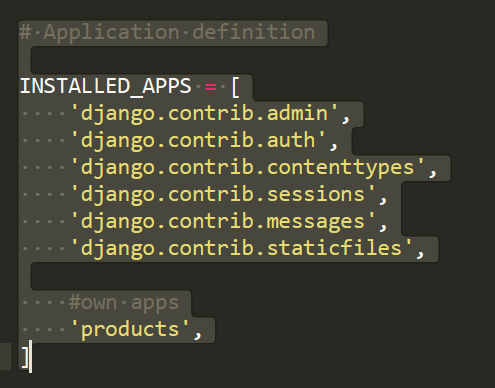
(*myDjango) C:\RISHI\Dev\myDjango\src>python manage.py startapp products*

Product app should do only product related things. It should be concise to product functionality. Go to models.py and edit this file with below content

first:



That’s it. Now go to settings and add it in Installed Apps.



**Every time you change a model or create a new model, run below commands**

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py makemigrations*

*Migrations for 'products':*

*products\migrations\0001\_initial.py*

*- Create model Product*

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py migrate*

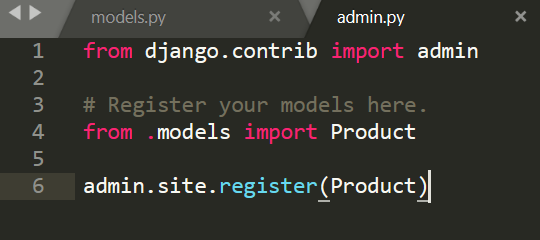
*Apply all migrations: admin, auth, contenttypes, products, sessions*

*Running migrations:*

*Applying products.0001\_initial... OK*

*(myDjango) C:\RISHI\Dev\myDjango\src>*

**Now go to admin.py and make these changes:**



Now start the server if its not already running and open Admin page.

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py runserver*

*Performing system checks...*

*System check identified no issues (0 silenced).*

*August 01, 2019 - 10:49:38*

*Django version 2.0.7, using settings 'myDjango.settings'*

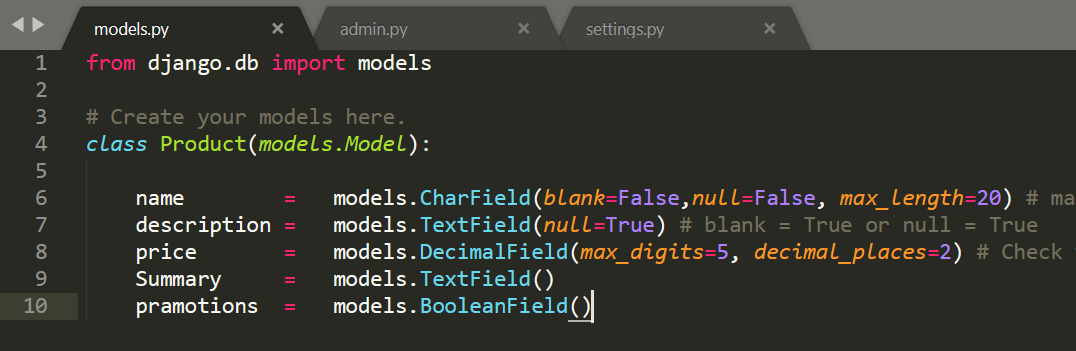
*Starting development server at http://127.0.0.1:8000/*

*Quit the server with CTRL-BREAK.*

Login to Admin page and see new Product got added on Page. Can you add new products? Check it out.

# Let’s change the Model:

We will use these model fields <https://docs.djangoproject.com/en/2.2/ref/models/fields/>



You may not need to delete DB, please try the steps directly:

*(septDjango) C:\RISHI\Dev\septDjango\src>python manage.py makemigrations*

*You are trying to add a non-nullable field 'promoted' to product without a default; we can't do that (the database needs something to populate existing rows).*

*Please select a fix:*

*1) Provide a one-off default now (will be set on all existing rows with a null value for this column)*

*2) Quit, and let me add a default in models.py*

*Select an option: 1*

*Please enter the default value now, as valid Python*

*The datetime and django.utils.timezone modules are available, so you can do e.g. timezone.now*

*Type 'exit' to exit this prompt*

*>>> False*

*Migrations for 'products':*

*products\migrations\0002\_auto\_20190910\_2055.py*

*- Add field promoted to product*

*- Add field summary to product*

*- Alter field description on product*

*- Alter field price on product*

*- Alter field title on product*

*(septDjango) C:\RISHI\Dev\septDjango\src>python manage.py migrate*

*Operations to perform:*

*Apply all migrations: admin, auth, contenttypes, products, sessions*

*Running migrations:*

*Applying products.0002\_auto\_20190910\_2055... OK*

Use this only when above options are not working.

If above commands doesn’t work.. then do below steps:

1. Stop the server
2. Delete all the files in migrations folder
3. Delete pyCache. You can keep init.
4. Delete sql.lite DB file

Lets make the change in Model now.:

Now before making migrations, you need to create user, you need to recreate it.. why? Remember you have deleted the database.

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py makemigrations*

*Migrations for 'products':*

*products\migrations\0001\_initial.py*

*- Create model Product*

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py migrate*

*Operations to perform:*

*Apply all migrations: admin, auth, contenttypes, products, sessions*

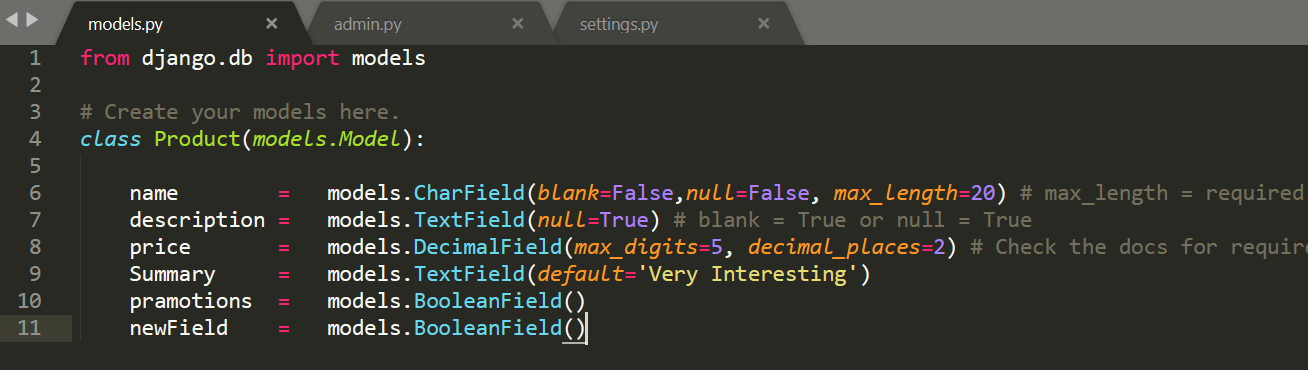
*Running migrations:*

*Applying products.0001\_initial... OK*

Now login to admin 🡪 navigate to create Product and enjoy the new view. Have you need Boolean Field?

# Handling Runtime Model Changes:

Make some changes in Model now:



Now, Database don’t know how to handle this new field. What should be the value of this field for previous entries. It will ask you same question while migrating.

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py makemigrations*

*You are trying to add a non-nullable field 'newField' to product without a default; we can't do that (the database needs something to populate existing rows).*

*Please select a fix:*

*1) Provide a one-off default now (will be set on all existing rows with a null value for this column)*

*2) Quit, and let me add a default in models.py*

*Select an option:*

Give option as you think. Option 2 is better when you want to go back and enter some default value. Here we will use option 1

*Select an option: 1*

*Please enter the default value now, as valid Python*

*The datetime and django.utils.timezone modules are available, so you can do e.g. timezone.now*

*Type 'exit' to exit this prompt*

*>>> True*

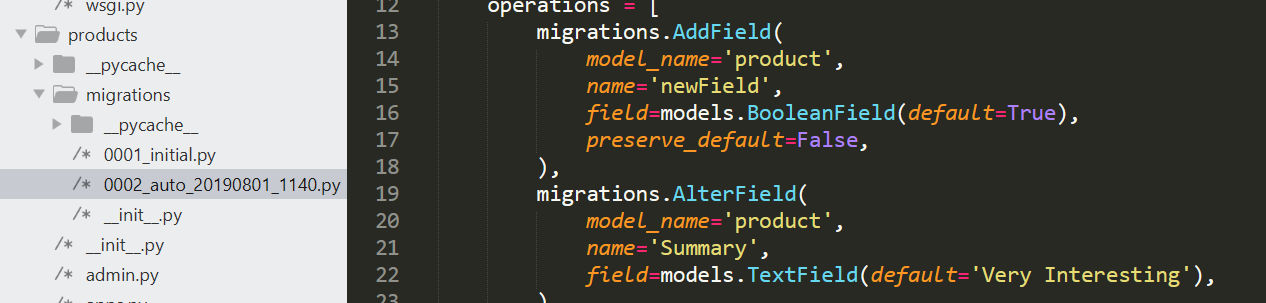
*Migrations for 'products':*

*products\migrations\0002\_auto\_20190801\_1140.py*

*- Add field newField to product*

*- Alter field Summary on product*

Observe migrations folder. Check how Django handled this migration.



Now, go ahead and add migrate, then start server and check previous products.

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py migrate*

*Operations to perform:*

*Apply all migrations: admin, auth, contenttypes, products, sessions*

*Running migrations:*

*Applying products.0002\_auto\_20190801\_1140... OK*

*(myDjango) C:\RISHI\Dev\myDjango\src>python manage.py runserver*

*Performing system checks...*

*System check identified no issues (0 silenced).*

*August 01, 2019 - 11:43:27*

*Django version 2.0.7, using settings 'myDjango.settings'*

*Starting development server at http://127.0.0.1:8000/*

*Quit the server with CTRL-BREAK*

blank = True or null = True? Which one to use**.**

Blank is about Field on Web Page. Null Is about database. If Blank is false, you cannot leave the field blank on Web Page.

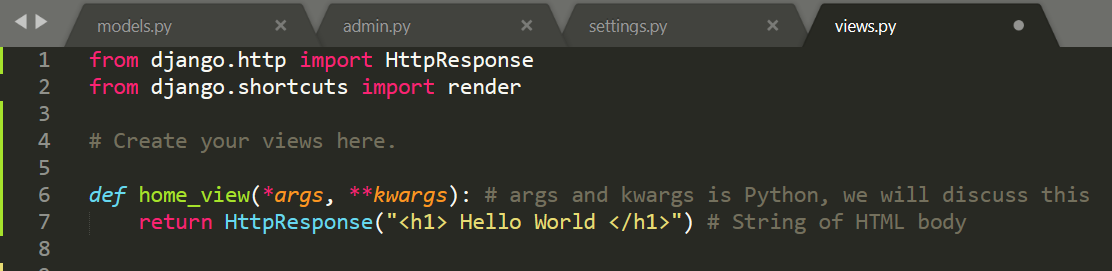
# Default Home Page to Custom Home Page

Before starting this section, just do this for me:

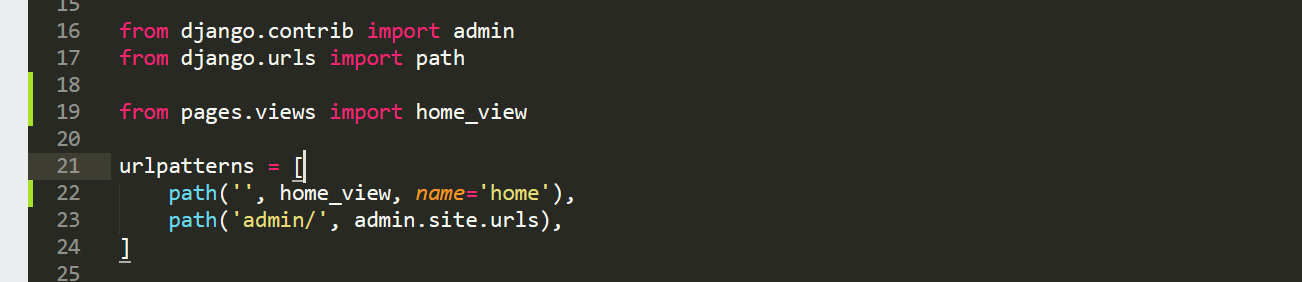
1. Create a new app called “pages”
2. Add your new module in settings.

If you don’t remember how to do it, please revise the videos.

1. Go to pages app and open views.py. Add below code.

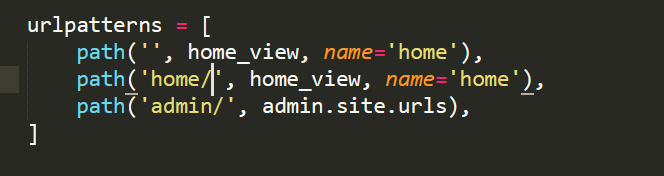


1. Now go to myDjango folder and open url.py – as mentioned in comments above, please add our view here



1. Now hit the URL: <http://localhost:8000/>

# URL Routing and Requests:



Try access URL with <http://localhost:8000/home/> 🡨 Observe if anything changes?

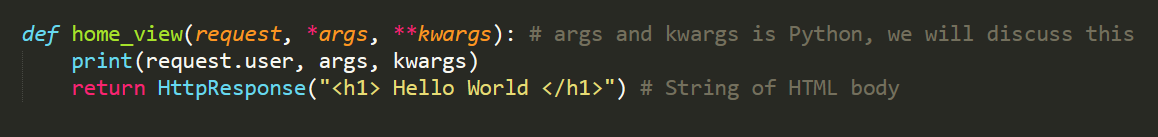
**Assignment:**

Can you add Contact page like this? Lets do it. Add as many view as you want. Play around with it.

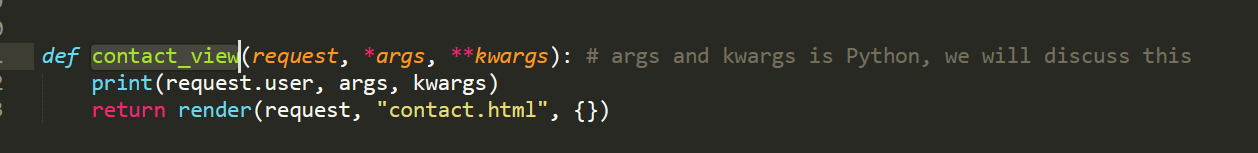


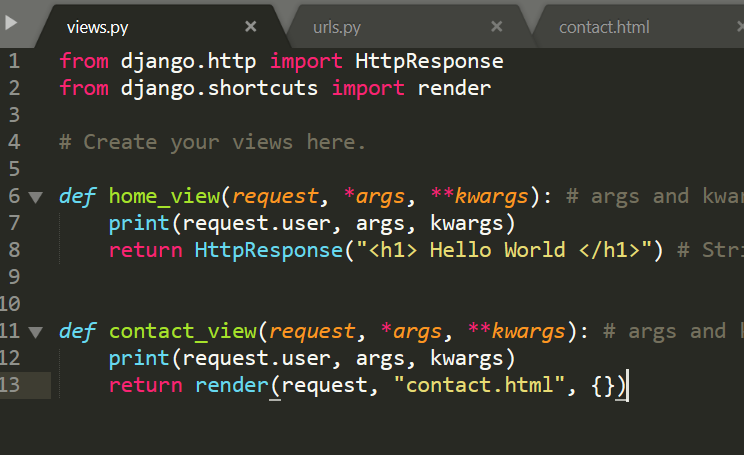
# What else we can do with views?

Can we access Request Parameters coming in web request.

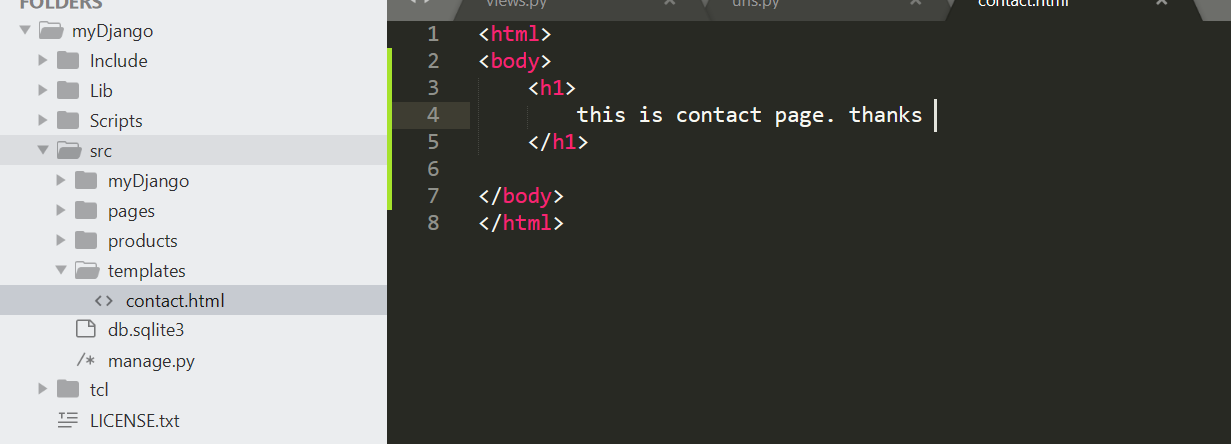


Lets go to HTML Page:

add method to views.py. Create HTML page



HTML Page in template folder. Create ***templates*** folder directly under src.

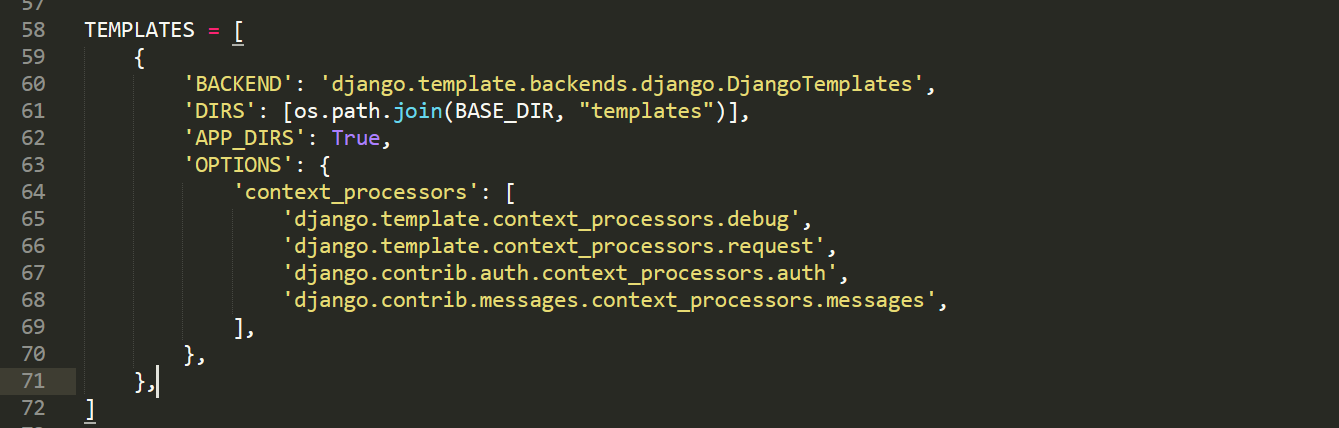


Go to settings and add this directory in DIRS in TEMPLATES section.



Run this project now. See if you get web page.

But the path given here will not work on your machine if I send you this code. So what should we do? Lets use BASE\_DIR

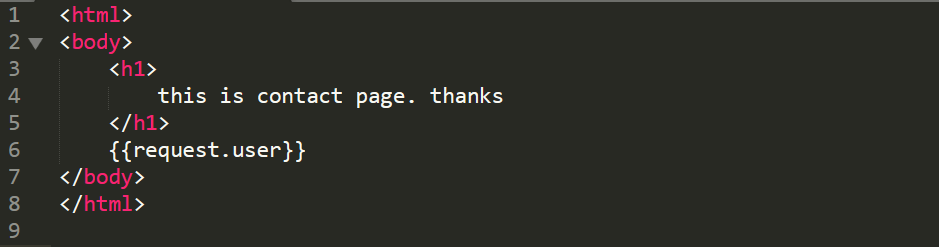


**Assignment 2:**

Can you create more html pages now like – about.html, contact.html, home.html and point it to web address? Please try and let me know.

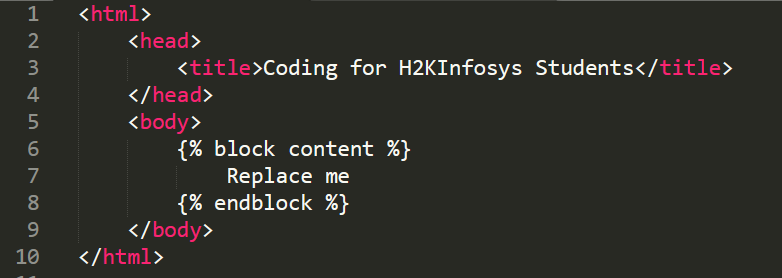
# Django Template Inheritance.

You can use UserModel in template engine. Lets try this for a simple thing like **request.user**.

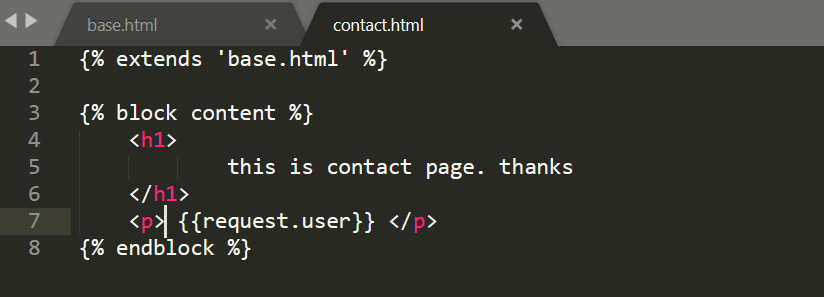


Check the response now.

You might need to use something common across all html pages, lets say a navigation bar or some meta data. So lets create a page called “**base.html”**  in template to handle this.



Now use this block content in contact.html:



Now refresh the output screen and check the output. Can you make similar change in other html pages you created?

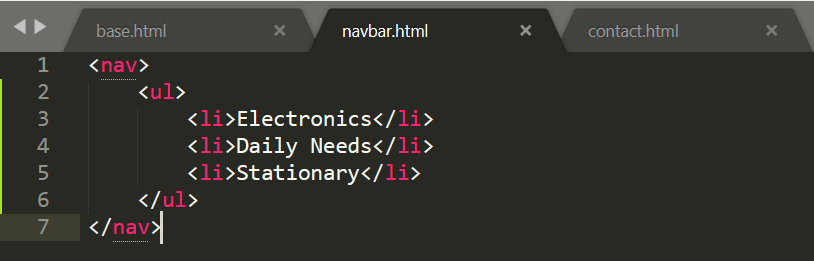
**Assignment 3:**

*Remember {% block content %} 🡪 block is Django stuff while “content” is my variable. You can change that too. Just make sure you are using same variable everywhere.*

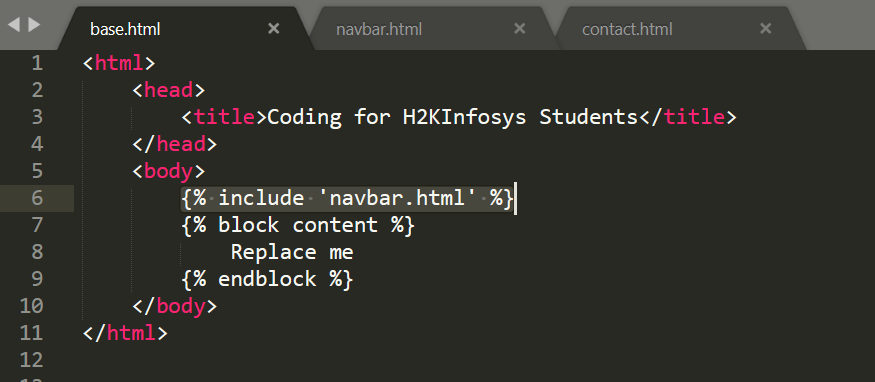
*Wait.. can you add more blocks then? Answer is – YES. Can you try that? Please do.*

# Include template tag:

Now I want a navbar in all pages. So do I have to add it in my base.html? YES. But that make base.html really heavy in UI related entities. So lets do this: I am creating another html page called navbar.html



Now use **include**  to use this html page in base.html

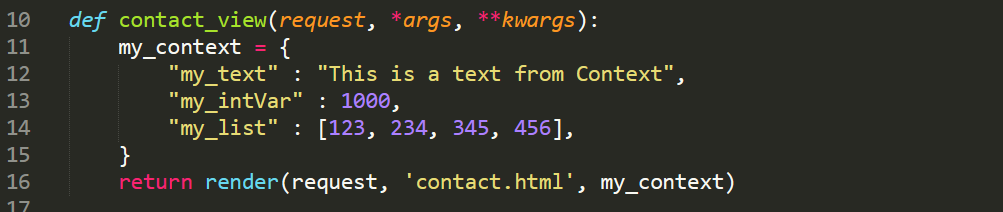


Load **contact.html** now. See the change.

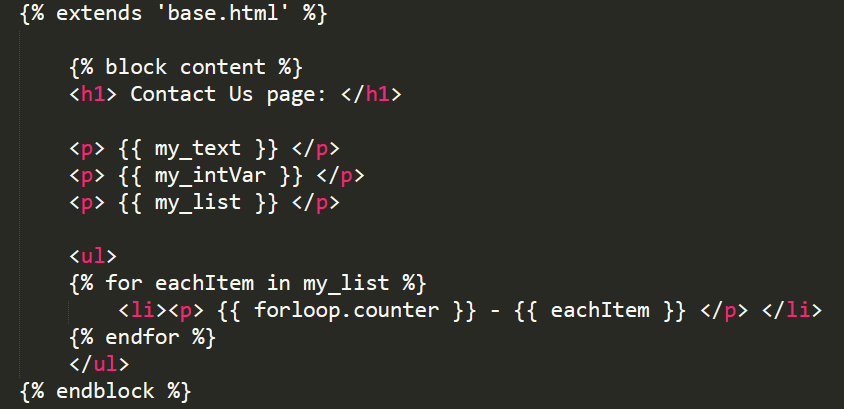
# Rendering Context in a Template:

What we really need on web page is Data from database isn’t it?

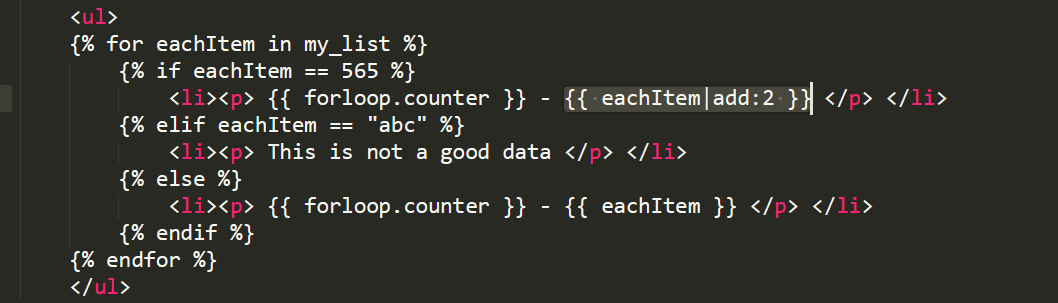
For Django, **User Page = template + context**. What do I mean by that? Remember that empty dictionary we passed to html page? Add something in it now.



Can we add a List in context? How can we show that on screen as HTML List?



Can we use **Conditions** in Templates?



What is this |add:2 🡨 this is built in template tag filter. Lets check Django page for this. Search for “Built-in filter reference”

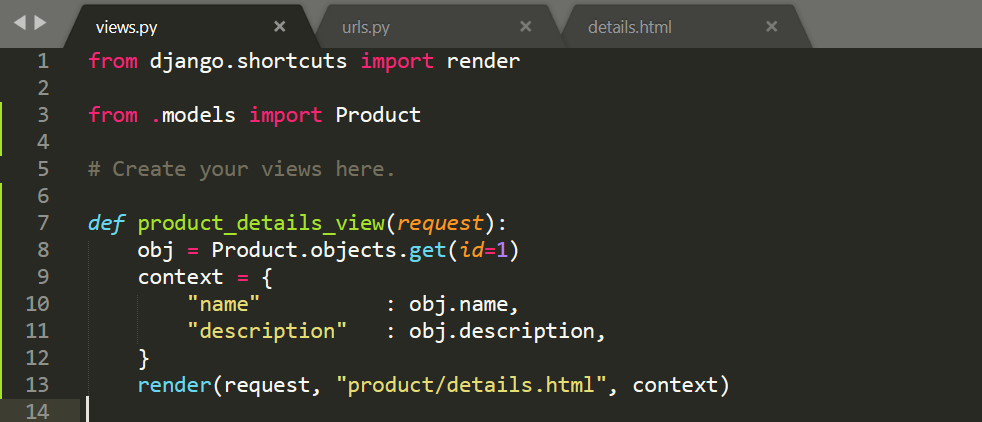
<https://docs.djangoproject.com/en/3.2/ref/templates/builtins/#add>

I strongly recommend you go through entire set and try few options.

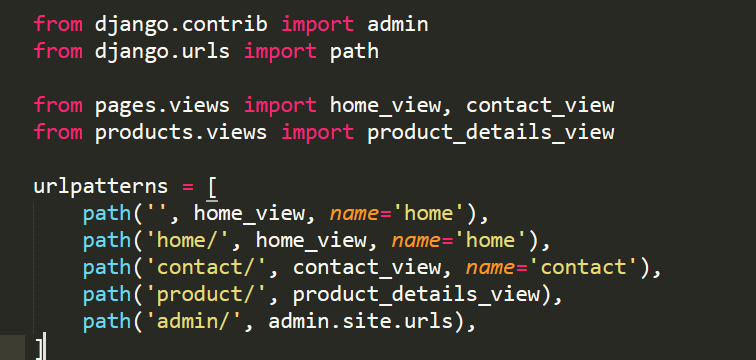
# Getting Data from DB:

Step 1: Open views from product app and create a view method. You access Product object with

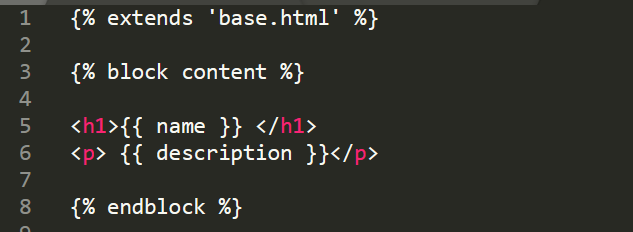
**Product.objects.get(id=N)**



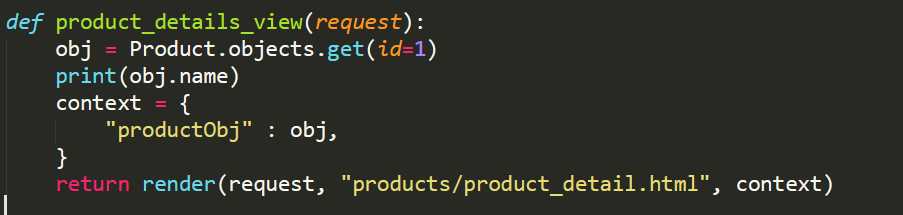
**Add this view in URLs.**



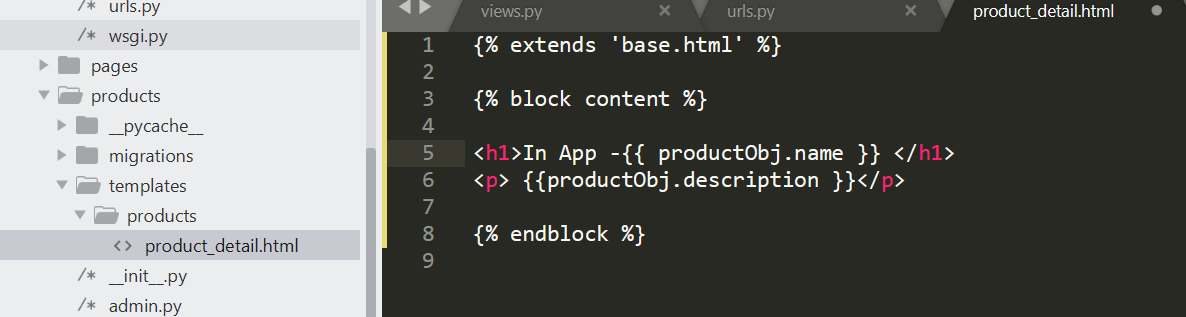
**Create product/details.html page in templates folder.**



**But then, why are we adding such a complicated context. Can we make it simple?**



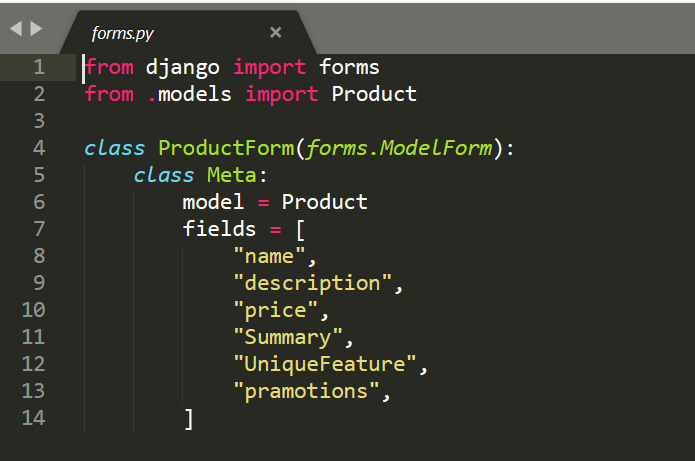
Now I am moving my template into App itself. How? Lets create template folder under product app.



This make my code more modular. Isn’t it?

# Django Forms:

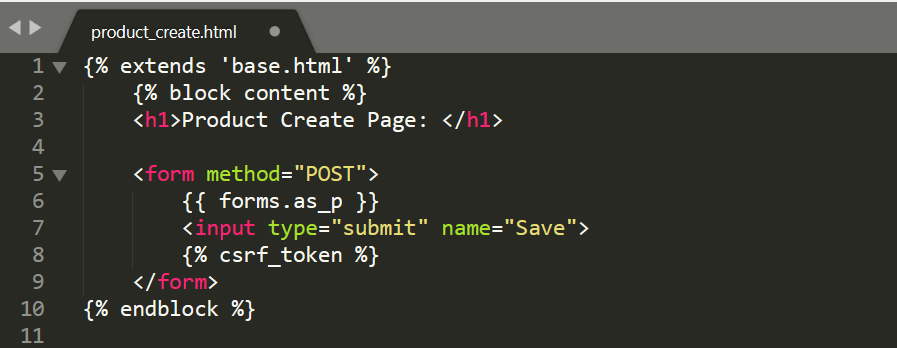
Aim is to create Product Form, which will take user inputs to create record in DB. First Step to create forms.py and create model Form like below:



We have to load this form from view. So add method:



Now complete product\_create.html



Understand csrf\_token as a part of security. We will discuss it in detail in class. Also, forms.as\_p makes entire form as <P> tags.

Next step? Add this view in URL.

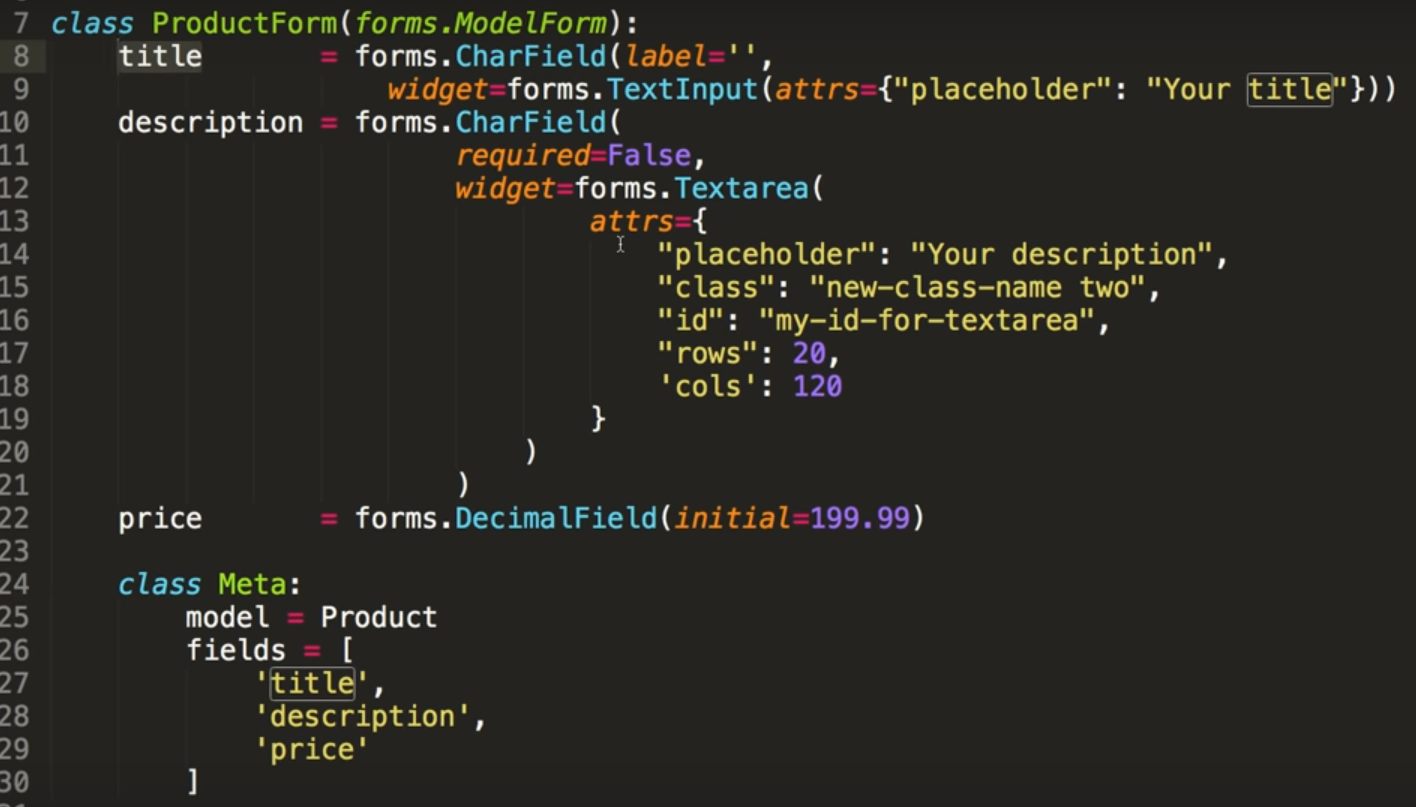


Restart the server and check the code. If you see any error, revise the steps and look for mistake. Still can’t get it? Email me: [rishi.h2kinfosys@gmail.com](mailto:rishi.h2kinfosys@gmail.com)

# Form Validation Method:

First I want to tell you that you can override every field in model class. Can we try one?

<https://docs.djangoproject.com/en/3.0/ref/forms/widgets/>

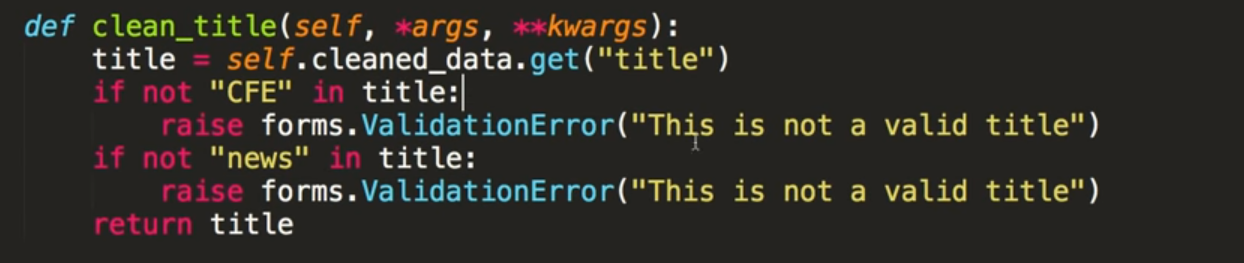


Lets talk about widgets and attributes.

With this code, you have overridden the form coming from Model by default.

# Now how validation works?

Suppose I want to check my title field has “H2K” letters, I can write a specific method with name **clean\_<my\_field\_name>(self)**



**Of course, if title is not valid, you can raise a ValidationError which will be shown on screen.**

**Assignment: Add a new field in Product Form as Vendor\_Email – email\_field and added in form. You should do basic email validation**

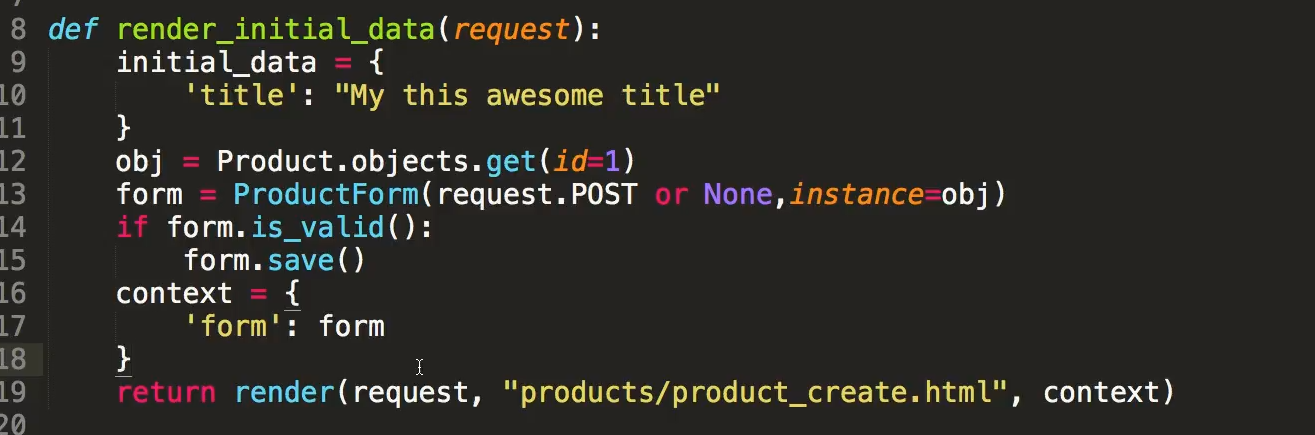
**So will you write email validation for me?**

# Setting Initial Data to Fields:

Use initial\_data dictionary to set initial data. Please see the example below:



You can also load Database entry to set as initial value:

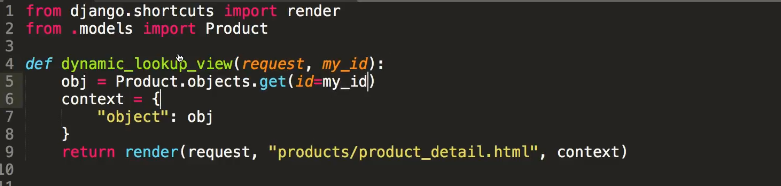


# Dynamic Data Loading:

Dynamically pass the data with URL:



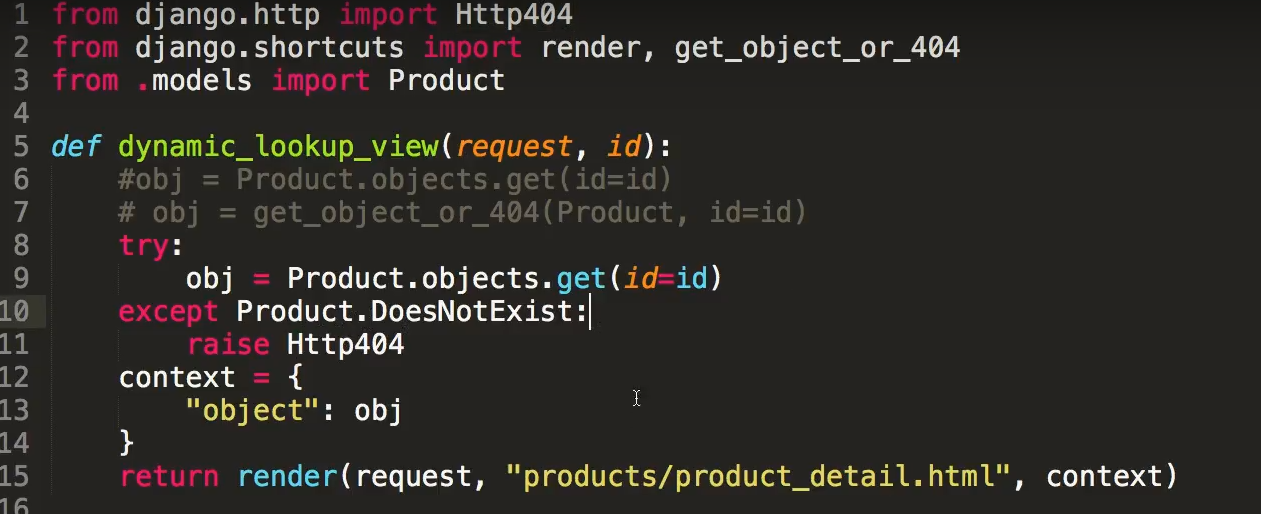
Handle this in method which shows view:



But what if someone sends an ID which doesn’t exists in DB?

# Handle DoesNotExist / 404

You can achieve 404 Handling with get\_object\_or\_404() or Http404 – see the code below:



# Switching Django Application to MySQL DB:

1. Install mysqlclient with pip

**C:\RISHI\H2K\aprDjango>pip install mysqlclient**

Collecting mysqlclient

Downloading mysqlclient-2.0.3-cp39-cp39-win\_amd64.whl (179 kB)

|████████████████████████████████| 179 kB 1.1 MB/s

Installing collected packages: mysqlclient

Successfully installed mysqlclient-2.0.3

1. Start MySQL Server and Create new schema for Django Project. (You can use existing schema also)

**CREATE SCHEMA `django\_app` ;**

1. Make the changes in Settings. Comment existing sqlite3 entry with below entry:

DATABASES = {

*'default'*: {

*'ENGINE'*: *'django.db.backends.mysql'*,

*'NAME'*: *'django\_app'*,

*'HOST'*: *'localhost'*,

*'PORT'*: *'3306'*,

*'USER'*: *'root'*,

*'PASSWORD'*: *'password'*,

}

}

1. Make Migrations & Migrate:

**C:\RISHI\H2K\aprDjango>python manage.py migrate**

No changes detected

**C:\RISHI\H2K\aprDjango>python manage.py migrate**

Operations to perform:

Apply all migrations: admin, auth, contenttypes, products, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0006\_require\_contenttypes\_0002... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying auth.0010\_alter\_group\_name\_max\_length... OK

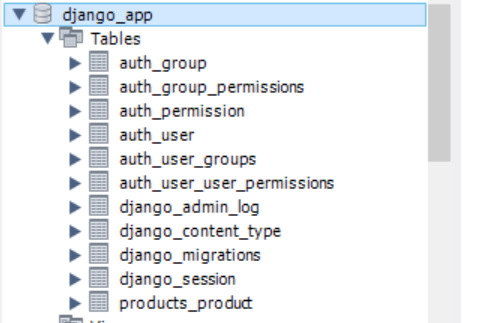
Applying auth.0011\_update\_proxy\_permissions... OK

Applying products.0001\_initial... OK

Applying products.0002\_auto\_20210416\_0615... OK

Applying sessions.0001\_initial... OK

1. Observe the tables getting created in Schema:



1. Create super user as old one will not work with new Database:

**C:\RISHI\H2K\aprDjango>python manage.py createsuperuser**

*Username (leave blank to use 'rishi'): rishi\_h2k*

*Email address: rishi.h2kinfosys@gmail.com*

*Password:*

*Password (again):*

*The password is too similar to the username.*

*Bypass password validation and create user anyway? [y/N]: y*

*Superuser created successfully.*

1. Start the Server

**C:\RISHI\H2K\workspace\aprDjango>python manage.py runserver**

*Watching for file changes with StatReloader*

*Performing system checks...*

*System check identified no issues (0 silenced).*

*April 21, 2021 - 18:29:45*

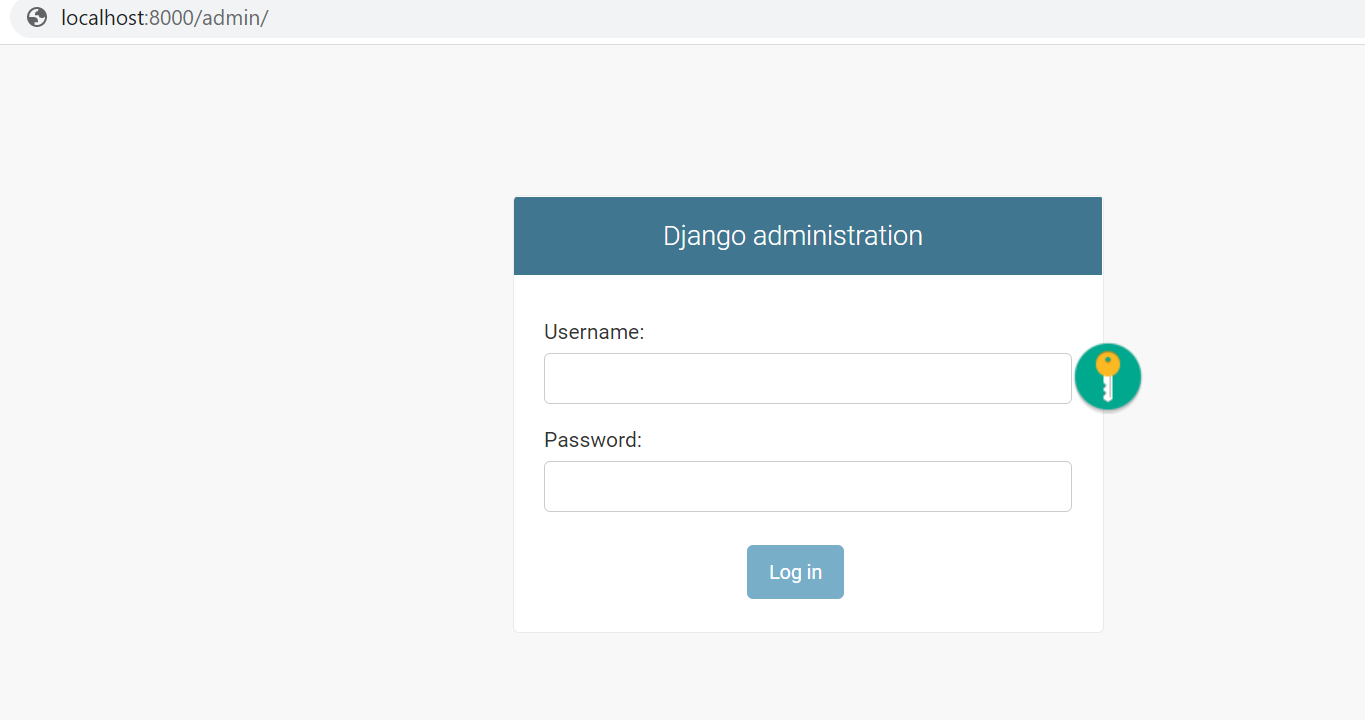
*Django version 3.0, using settings 'aprDjango.settings'*

*Starting development server at http://127.0.0.1:8000/*

*Quit the server with CTRL-BREAK.*

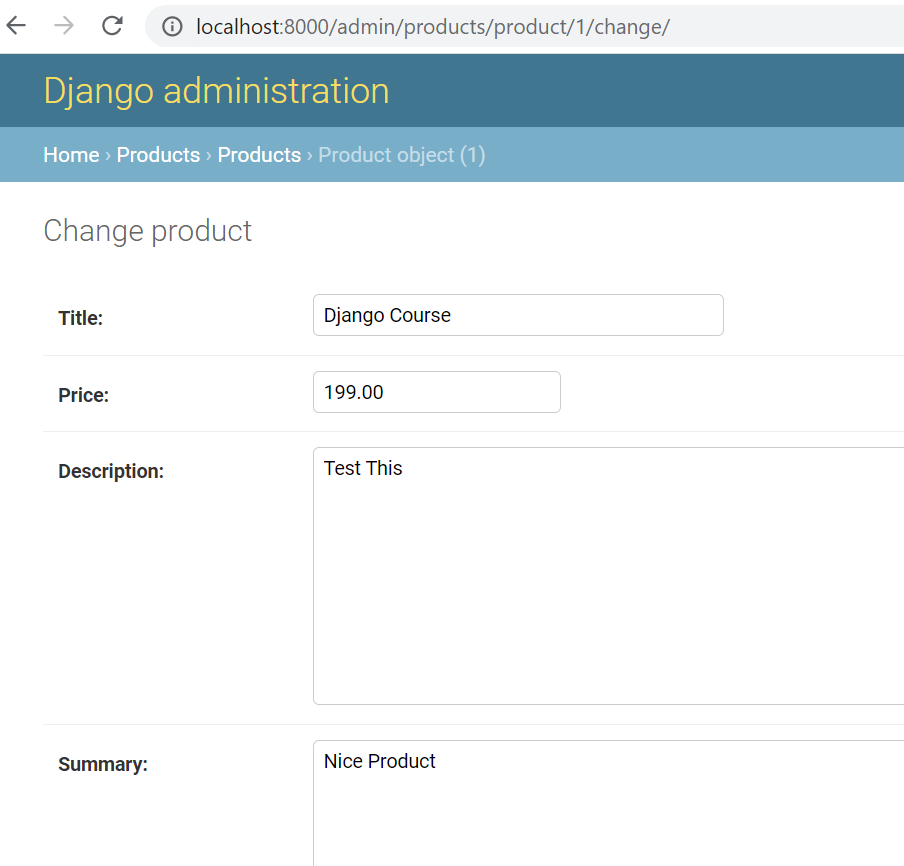
1. Now test admin console – all other components of applications can be tested now.

<http://localhost:8000/admin/>





1. Add New Product and query the database:



MySQL View:

