**Re-Migration**

Earlier using pgadmin and postgres we were able to create the tables in Django without need to write a single line of sql query to create tables. We know that the tables are empty.

Graphical user interface, application

Description automatically generated

Why do we need to know about remigration? – for example there is a mistake in our code and we missed to push one field to our postgres database we might have to do a migration again, another case is we want to add some new field based on the new business requirement.

Let’s take a look at how we re-migrate

To make use learn from the mistakes creator of this content Mr.Naveen Reddy has missed to add () after the IntegerField hence if you observe the destination database tables we cannot see price column.

A screenshot of a computer

Description automatically generated with medium confidence

You can add the field using sql query in admin panel or follow the below steps.

Let’s add a () and perform the steps below to fix it.

Text

Description automatically generated

Run the below commands to perform the migration and select the default values, keeping in mind that we do not have any rows in the table.

A person speaking into a microphone

Description automatically generated with low confidence

To complete the migration we need to run migrate as shown below. Unlike in the screenshot we will get that output that these columns are updated.

Text

Description automatically generated

If you check/refresh the database should see that the price column is updated.

Graphical user interface, application

Description automatically generated

We can also see that there are 2 files 1)Initial migration and 2)destination price migration

A screenshot of a computer

Description automatically generated with medium confidence

**Django Admin panel**

Till now we have created a database and tables in it, without DATA. Yes, we want to push some data to the database. We can do it in 2 ways 1)by creating a page where a user can push data or, 2) we can have an admin page.

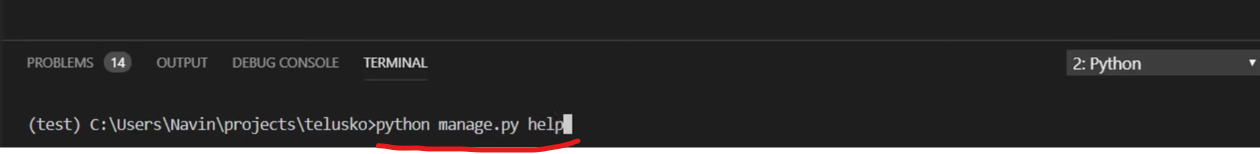
Normally admin page can be accessed by the admins or superusers. As a user you can push data to a facebook account. Here we are working with destination page, this data will be pushed by the agents or admins. Let’s access the admin page which we have it ready for us, there is no need to create an admin page. Just access your site url/**admin** this will render the admin page as shown below.

Graphical user interface, application

Description automatically generated

It will ask for username and pwd, we don’t have one for this site as of now. Since the creator of this content is using the Django he got his creds picked up automatically, however we will have to face the empty fields when we access it for the first time.

We need to create a super user – to create super user, let’s jump to our IDE and try help so that we can see the options available for us.



Is there something that’s helpful for our task of creating a new superuser? – observe the help options runserver, collectstatic, migration related and somewhere we can see that ‘createsuperuser’ command available.

Graphical user interface, text

Description automatically generated

Let’s run the command and it will ask for username and password as highlighted below: provide the details accordingly and proceed further.

Text

Description automatically generated

In our admin page let’s login with the credentials we had created.

Graphical user interface, application

Description automatically generated

We can access the administration portal as shown below. Click on users and we can see that we have user that we have logged in and other users if you have created any. Notice that we also have an options for groups

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application

Description automatically generated

Click on the user and we can update the profile as well.

Let’s come back to our main objective of adding the destinations where we want to work with databases.

Let’s go back to our IDE and we can notice that there is **admin.py** and here we need to register the model and automatically it will give you the page. In travello folder admin.py perform the below changes

A screenshot of a computer

Description automatically generated with medium confidence

After the previous update, go back to the admin panel and refresh the browser, we can notice that there is Destination as highlighted below.

Graphical user interface

Description automatically generated

When you click on destination we can see that there are no destinations at the moment. We can add destination as highlighted below.

A picture containing background pattern

Description automatically generated

When you click on add destination we can see the below form, the images ‘choose file’ will be coming from the users as they are passing images. Earlier we had worked with static resources where we had a different folder ‘assets’ but this time we have to go for a different folder because this will be users data. We will look at how to implement it in the next video.

Graphical user interface, text, application, email

Description automatically generated

How to work with media, how to push the media content from here in the media folder and how to access it?