**Postgres and PgAdmin setup**

Earlier we have seen how to pass data to views but then from where we will get this data and that’s where we want a database. How will we create a database? If you want to create a database we need a database engine. There are various RDBMS available like Oracle, MySQL, PostgreSQL, IBM DB2.

Since we are working with Python we are going to use PostgreSQL

Logo, company name

Description automatically generated

Let’s go to our search engine google and try to find it. Remember always make use of official downloads for our own good. As a security student I can give reasons why it is recommended to use official downloads but that will be out of scope of this topic. However, if you want to know more about it feel free to reach me or feel free to do your own research.

Graphical user interface, text, application, email

Description automatically generated

Now since we found our official site, based on our operating system let’s download the associated software using the link below.

Graphical user interface, text, application, email

Description automatically generated

When you install postgreSQL, it will give you database for sure. We also need a UI using which you can use it. **UI – User Interface** **using which you can use or interact with the software.** And our UI is PgAdmin and download it according to your operating system.

Graphical user interface, text, application, email

Description automatically generated

Now install the PostgreSQL first leaving the default settings, providing the password for user **postgres** which is important. Passwords are always important 😊 so make sure you handle them safely and try not to forget them, if you forget or mess up with them “welcome to the world of service denial” 😊 maybe for sometime atleast until you figure it out.

Graphical user interface, application

Description automatically generated

We will get a port number **5432** on which PostgreSQL runs and proceed with the default settings and finish the installation. Below screen gives the confirmation that PostgreSQL is completely installed.

Graphical user interface, application

Description automatically generated

Now if we want to access it we need one more software pgadmin. Let’s install the pgadmin software with all the default settings. Give it sometime as it will take a while to get it installed. We will see the below screen after successful installation of pgadmin

Graphical user interface, application

Description automatically generated

Launch it and we will see the pgadmin as shown below on our browser.

Graphical user interface, text, application, email

Description automatically generated

Give the password accessing the servers as highlighted on the left and proceed further. Now w should enter the PostgreSQL password which we used earlier.

Graphical user interface, application

Description automatically generated

After giving the password we will see the screen as shown below.

Graphical user interface, application

Description automatically generated

We can see that we have some databases available here, by default we have postgres database as shown below. We can create our own databases later

Graphical user interface, text, application, email

Description automatically generated

We can take a look at different options available and explore the schemas(tables) and we can see that we do not have any table at the moment. We want to create a table so that we can store some data in the database from the page which we are working with or we might want to fetch some data from here. We can write a database and tables using sql queries.

Graphical user interface, application

Description automatically generated

**Models and Migrations**

We are fetching the data from database and we are trying to connect it to the application. We will make use of the pgadmin which we have installed and access our postgres. Let’s create another database and how can we do it? – check the below screenshot.

Graphical user interface

Description automatically generated

Fill the form, give it a name and save it.

Graphical user interface, application

Description automatically generated

New database is created as shown below and if you observe the schemas, there is no table here:

Graphical user interface, text, email

Description automatically generated

To connect your application with this PostgreSQL we need to do some configuration and that configuration we will do it in our IDE i.e., VSCode

Navigate to settings.py of our main project and check for databases and make the below highlighted changes 1) engine: postgresql 2) name: telusko 3) USER 4) Password 5) HOST

Text

Description automatically generated

Postgres and Django are different softwares and how will they connect? – there is a connector in between which is shown below. Understand the project description atleast the purpose or function of it.

Graphical user interface, application

Description automatically generated

Install it using the command ‘pip install psycopg2’

Text

Description automatically generated

Now let’s jump to our ORM concept – it will create database tables for us. Ex: for authentication. If you have your own models it will create database. Before migration we need a model, the table which we want ‘destination’ depend upon your model.

We have a model file in travello where we had created a class, but the destination is a simple class and we need to convert this into a model. This will not work, so how do we do it? – update as highlighted below ***models.model***

When we are creating a table, table will have types of columsn as well ex: name (type would be varchar) that’s what we do in sql. We do not need to mention the id because in database it will automatically create a ***primary key*** for us. To know the fields we can check official documentation

<https://docs.djangoproject.com/en/4.0/ref/models/fields/>

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Now our models.py will look like this after updating

A screenshot of a computer

Description automatically generated with medium confidence

Now if you want to create a table we need to migrate our model to the database. To do so we need to run a command, we can see that we have a folder called migrations(empty) in our project folder.

Text

Description automatically generated

The command we will run is ‘python manage.py makemigrations’ as highlighted below.

Text

Description automatically generated

One more update to fix the error which came from before command, update our **settings.py** with the highlighted line.

Text

Description automatically generated

Run the command again and you should see an error again as shown below. Let’s fix it by running the **‘pip install pillow’ –** whenever you want to work with images in Django and since we want to upload them, we need to use pillow library.

Text

Description automatically generated

Run the command again and we should see it worked. DO NOT Forget to run the **pip install Pillow** before doing so, it will create an error else.

Graphical user interface, text

Description automatically generated

Now if we expand our migrations we can see that we have **001\_initial.py** and if we open that file we can see what migration we are doing.

A screenshot of a computer

Description automatically generated with medium confidence

We are able to see the migration we are going to perform : we can notice that even though we remove the ID it is automatically got added to the file.

Text

Description automatically generated

Till this point we have created a migration file BUT NOT the database yet.

Let’s run the command as highlighted below. Sqlmigrate travello(app) 0001(version/migration number)

We can see the sql query that it created for the table and ran for us. **Create table….**

Graphical user interface

Description automatically generated with low confidence

Let’s run the last command: to migrate the contents ***‘python manage.py migrate’***

Graphical user interface, website

Description automatically generated

Output will be something like this

Text

Description automatically generated

No errors.

Let’s check our browser refresh our database

A picture containing application

Description automatically generated

Graphical user interface, application

Description automatically generated

All the tables are given by Django, we don’t even need them as of now. We focus on the travello\_destination table highlighted above. Rightclick on table and show data will give the empty table as we have not entered data as of now. We have a structure as of now id,name,….

Graphical user interface, text, application

Description automatically generated

We got a table from a class – ORM