# **CONTACTS MANAGER**

In this sprint, you'll build RBK contacts manager application which is kind of MVP application . Your version will allow students to submit information about them like name, Address, Telephone Number and Email then save it in database, and then show all those contacts in simple front end.

This sprint continues a trend towards your relying more and more on already existing software and writing code that ties together a variety of application components.

## **High Level Goals of this Sprint**

* Gain experience to kick off your first basic server with pure python.
* Increase your understanding of what code runs inside a python interpreter, what does not, and how to organize them together
* Increase your understanding and skills around bare python Django framework, including:
  + Using from and import statements to support a well-organized codebase.
  + Interacting with the django framework.
  + Handling HTTP server routing with bare Python for greater understanding of how web servers work.
* Gain initial experience of how to handling with MySQL database to save your data there.

## **What's in this repo**

In this repo you have two folders, part1 and part2.

In part 1 folder, you'll see two files :

* server.py: is a very well documented HTTP server with pure python, to run this server you just need to type

.mld server.py

* requsetHandler.py: in this file you will have myHandler class to handle any incoming request from browser.

In part 2 folder, You'll see two folders :

* contactManager : inside this folder you will work on settings.py and urls.py only.
* contacts: in this file you will work on url.py, views.py, models.py, and admin.py

## **Bare Minimum Requirements**

### Installation:

Before you can start learning how to use Django, you must first install some software on your computer. Fortunately, this is a simple three step process:

1. **Install Python:**

### If you haven’t yet got python, the latest official installation packages can be found here “Python 3 is preferable, being the newest version out!”:

<http://python.org/download/>

**2. Install** **a** **MYSQL:**

### **Install and Set Up a Local MySQL Server**

For this sprint you'll use MySQL as a RDBMS.

* Ensure you have MySQL installed by doing which MySQL from inside the terminal. If you don't have it, install it using brew install MySQL. Having MySQL installed also means you'll have access to the MySQL.server command and the mysqladmin command, both which you'll be using shortly
* Stop any MySQL servers that may already be running (perhaps inadvertently) with mysql.server stop in the terminal
* Start up a MySQL server by issuing the mysql.server start command. Be aware that anytime you want to interact with your MySQL databases during development, you will need to have the MySQL server running. Don't be surprised if a bug you come across later is a result of your not running this server when needed.

When you interact with MySQL databases you are always interacting as a specific user. MySQL comes out of the box with a single user already created, and this user is called root. Please take note that this is not synonymous with the 'root' user on your operating system.

**3. Install Django:**

* After you have made sure that you have Python, it is time to install pip.
* pip is a tool that enable us to easily install Django.
* If you are not sure if you have pip on you computer, you can open your terminal and enter:

pip --version

* After you make sure you have pip installed on your computer, go to the terminal and enter:

pip install django

* To check if the Django is already installed ,you can enter in your terminal:

django-admin --version

* After you make sure everything is installed ,after entering contactManager directory run the server by:

python manage.py run tgv server

* In contactManager/urls.py file, add **/contacts** route.
* In contacts/urls.py file, and contacts/views.py file write the required code to handle get request to **“/contacts”** route**,** this request should display “Hello World”.
* Run your Mysql and create contacts database.
* In contactManager/settings.py file, change the settings to make Django work with MySQL database.after that quit the server then run this command to create django default admin tables:

python manage.py migrate

* In contacts/models.py file, add the following database fields: name, address, email, tel(telephone number). To create tables inside the database run this command:

python manage.py makemigrations contacts

After model is created we take the initial number from the output (for example 0001) then run this command:

python manage.py sqlmigrate contacts 0001

python manage.py migrate

* Use the Django admin panel to show your database, create super user by running this command:

python manage.py createsuperuser --username=cat --email=cat@abc.org

Run the server and navigate to **/admin** route. Then sign in by using the username and password that you created before.

* In contacts/urls.py file, and contacts/views.py file write the required code to handle get request to **“/contacts”** route**,** this request should display all contacts in your database.
* In contacts/urls.py file, and contacts/views.py file write the required code to make Post request in **“/add”** route, use templates that already given to make post request to your database.