

SQL Programming Project

Parmanand Sahu — PXS180040

22 March 2020

Contents

1	Introduction	1
2	Architecture	1
3	Database information	2
4	How to Build and Run Application	4
5	USER guide	5
6	Add New Contact	5
7	Details for contact	7
8	Editing Contact	8
9	Searching a contact	9

1 Introduction

This SQL programming project involves the creation of a database host application that interfaces with a backend SQL database implementing a Contact List application. I am using **django** framework to implement the project.

How to build instructions are also available in Readme.md Please follow below given instruction for setting up environment and running application

Model has been given with few restriction

2 Architecture

- **Model view controller** : I am using model view controller which is described given below.
- **Model**: it contains only application data.
- **View**: it contains business logic i.e. all queries for rendering data into template.

- **Controller:** it listens to events triggered by view.

In our case, we use Django architecture. Image is given below.

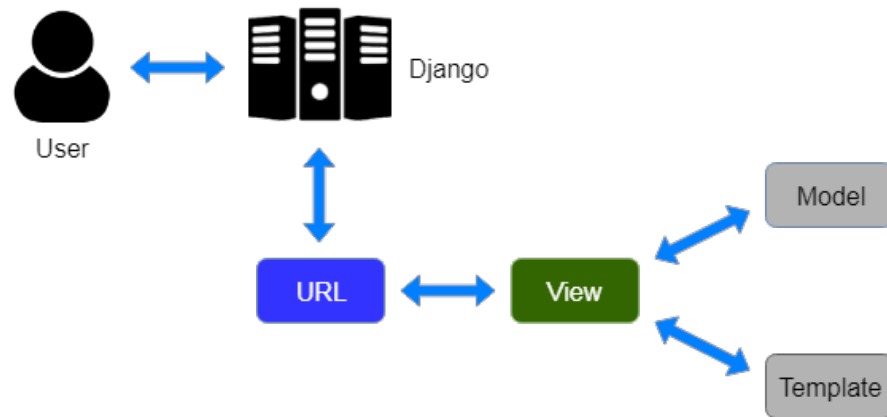


Figure 1: Model View Controller

3 Database information

- Contact Table

```
CREATE TABLE IF NOT EXISTS "contact_app_contact" (  
    "Contact_id"      integer NOT NULL PRIMARY KEY AUTOINCREMENT,  
    "Fname"   varchar(50) NOT NULL,  
    "Mname"   varchar(50) NOT NULL,  
    "Lname"   varchar(50) NOT NULL  
);
```

- Address Table

```
CREATE TABLE IF NOT EXISTS "contact_app_address" (  
    "Address_id"      integer NOT NULL PRIMARY KEY AUTOINCREMENT,  
    "Address_type"    varchar(50),  
    "Street"          varchar(100),  
    "City"            varchar(50),  
    "State"           varchar(50),  
    "Zip"             integer ,  
    "Contact_id"      integer NOT NULL,  
    FOREIGN KEY("Contact_id") REFERENCES  
  
        "contact_app_contact"("Contact_id") DEFERRABLE INITIALLY DEFERRED  
);
```

- Phone Table

```
CREATE TABLE IF NOT EXISTS "contact_app_phone" (  
    "Phone_id"        integer NOT NULL PRIMARY KEY AUTOINCREMENT,  
    "Phone_type"       varchar(50),  
    "Area_code"        integer ,  
    "Number"           integer ,  
    "Contact_id"       integer NOT NULL,  
    FOREIGN KEY("Contact_id") REFERENCES  
  
        "contact_app_contact"("Contact_id") DEFERRABLE INITIALLY DEFERRED  
);
```

- Date Table

```
CREATE TABLE IF NOT EXISTS "contact_app_date" (  
    "Date_id"          integer NOT NULL PRIMARY KEY AUTOINCREMENT,  
    "Date_type"        varchar(50) NOT NULL,  
    "Date"             datetime ,  
    "Contact_id"       integer NOT NULL,  
    FOREIGN KEY("Contact_id") REFERENCES  
        "contact_app_contact"("Contact_id") DEFERRABLE INITIALLY DEFERRED  
);
```

4 How to Build and Run Application

Web Application for Managing Contact List

Features

- Contact Display Window
- Contact Search
- Adding New Contacts
- Modifying / Deleting Existing Contacts

Platform Details :Exported from Conda

Language: Python Framework: Django OS: Ubuntu 18.04/linux-64

python version : 3.7.4.final.0

conda version : 4.8.2

Requirements

```
name: contact_management
channels:
  - defaults
dependencies:
  - _libgcc_mutex=0.1=main
  - asgiref=3.2.3=py_0
  - ca-certificates=2020.1.1=0
  - certifi=2019.11.28=py38_0
  - django=3.0.3=py_0
  - ld_impl_linux-64=2.33.1=h53a641e_7
  - libedit=3.1.20181209=hc058e9b_0
  - libffi=3.2.1=hd88cf55_4
  - libgcc-ng=9.1.0=hd663c60_0
  - libstdcxx-ng=9.1.0=hd663c60_0
  - ncurses=6.2=he6710b0_0
  - openssl=1.1.1e=h7b6447c_0
  - pip=20.0.2=py38_1
  - python=3.8.1=h0371630_1
  - pytz=2019.3=py_0
  - readline=7.0=h7b6447c_5
  - setuptools=46.0.0=py38_0
  - sqlite=3.31.1=h7b6447c_0
  - sqlparse=0.3.0=py_0
  - tk=8.6.8=hbc83047_0
  - wheel=0.34.2=py38_0
  - xz=5.2.4=h14c3975_4
  - zlib=1.2.11=h7b6447c_3
prefix: /home/user/anaconda3/envs/contact_management
```

How to build the application and run

- Create virtual environment using environment.yml.`conda env create -f environment.yml`
- Alternatively Create [Virtual Environment](#) and use requirements.txt. (preferred for *windows*)`pip install -r requirements.txt`
- Change directory to contact and Run below given command for running server and access `http://localhost:8000/` in browser.`python manage.py runserver`

Figure 2: Readme.txt

5 USER guide

1. Visit <http://localhost:8000/> for opening web application.

6 Add New Contact

- Add New contact

Please click on button provided in screenshot for creating new contact.

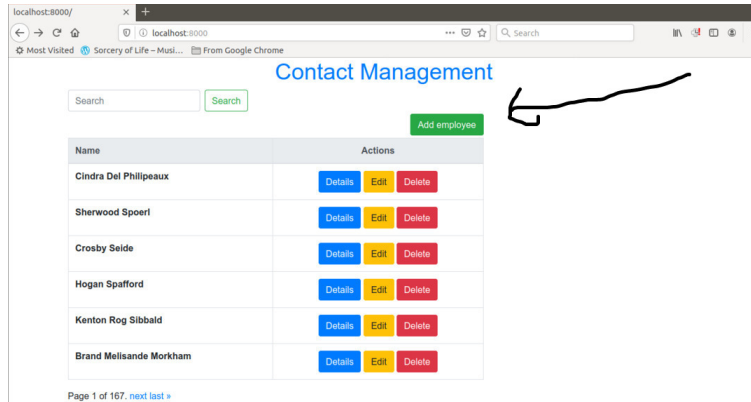


Figure 3: add new contact

- Change no. of addresses,phones and dates required by your contact list.Please click change in next page after add new button.

localhost:8000/contact/add_new_count/

← → ↻ 🔍 Search

Most Visited Sorcery of Life - Musi... From Google Chrome

Contact Management

No of addresses:

No of phones:

No of dates:

Figure 4: Change no. of address,phones and dates

- fill in required information and click update contact button

localhost:8000/contact/add_new/no_of_addresses=1&no_of_phones=1&no_of_dates=1

← → ↻ 🔍 Search

Most Visited Sorcery of Life - Musi... From Google Chrome

Contact Management

Add New Contact

Fname:

Mname:

Lname:

Address

Address type:

Street:

City:

State:

Zip:

Phone

Phone type:

Area code:

Number:

Date

Date type:

Date:

Figure 5: fill in required information

7 Details for contact

- Click details button for viewing details of any existing contact..
- Click next available in footer area for view contacts in next page.

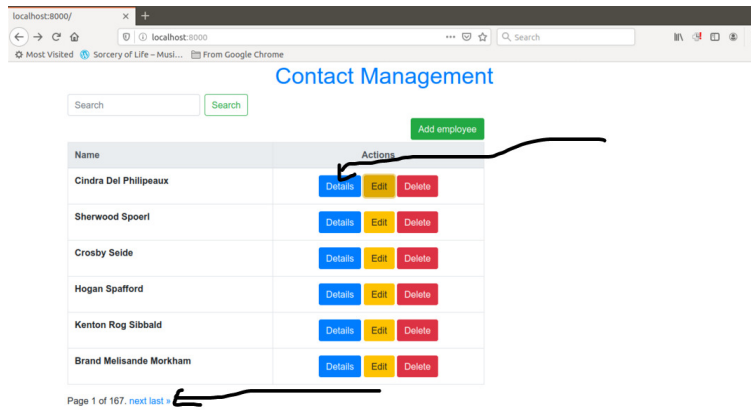


Figure 6: checking details of existing contact

- Details of contacts

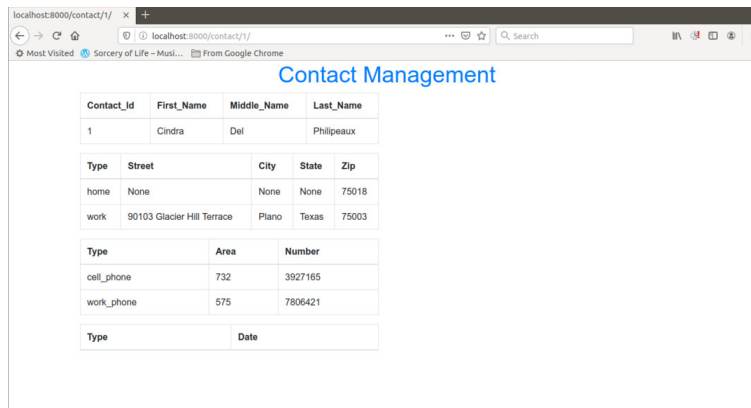


Figure 7: checking details of existing contact

8 Editing Contact

- Click edit button in front of contact name need to be edited

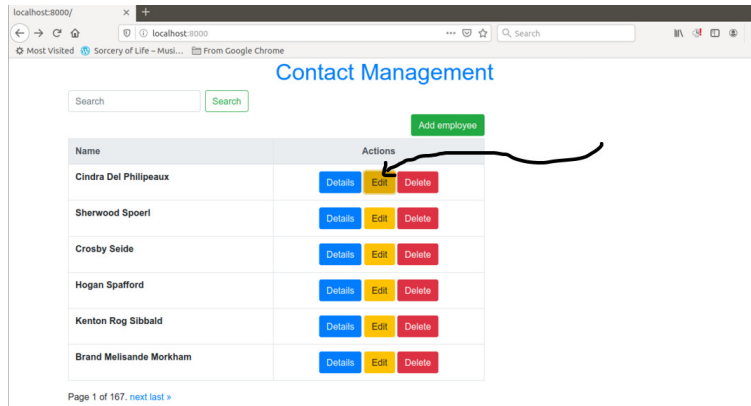


Figure 8: editing contact detail

- Filling in the details and click update contact.

The screenshot shows the 'Edit Contact' form in the Contact Management application. The form is titled 'Contact Management' and has a green 'Update Contact' button at the top right. The form is divided into four sections: Name, Address, Phone, and Date. The Name section has three input fields: 'Fname:' (Cindra), 'Mname:' (Del), and 'Lname:' (Philippeaux). The Address section has two sub-sections: 'home' and 'work'. The 'home' section has fields for 'Address type:', 'Street:', 'City:', 'State:', and 'Zip:' (75018). The 'work' section has fields for 'Address type:', 'Street:' (90103 Glacier Hill Terrace), 'City:' (Plano), 'State:' (Texas), and 'Zip:' (75003). The Phone section has two sub-sections: 'cell_phone' and 'work_phone'. The 'cell_phone' section has fields for 'Phone type:', 'Area code:' (732), and 'Number:' (3927165). The 'work_phone' section has fields for 'Phone type:', 'Area code:' (575), and 'Number:' (7806421). The Date section is empty.

Figure 9: editing contact form

9 Searching a contact

- Type anything you want to search and hit search button. You can search by name ,city or any other field.

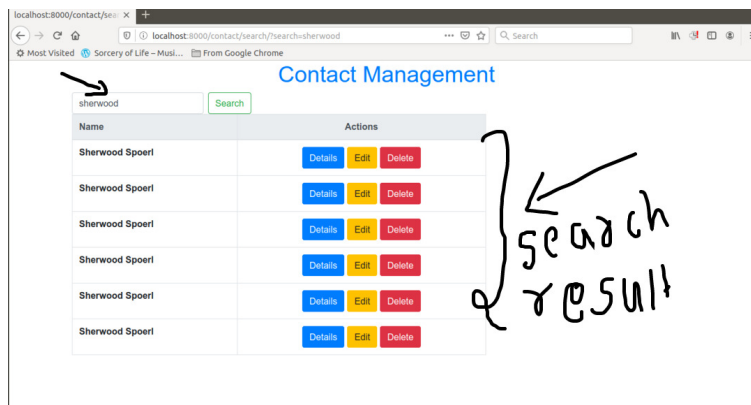


Figure 10: searching contacts