Documentation

T Sanjev Vishnu

April 14, 2020

1 Introduction

This is the documentation of assignment for web nominee. This invloves making a reimbursement portal.

My approach for implementing this portal is to make a django app since this framework is extremely useful in cases like this and also our website runs on django.

2 Instructions to Run

Before running the code, make sure you have a stable internet connection and python3, django installed

Download the repository and navigate to the directory in which you have downloaded the repository

Open terminal and run the following code

```
$ cd itsp-reimbursement-portal/itsp        #to get into the folder itsp
$ python3 manage.py runserver              #creates a development server
```

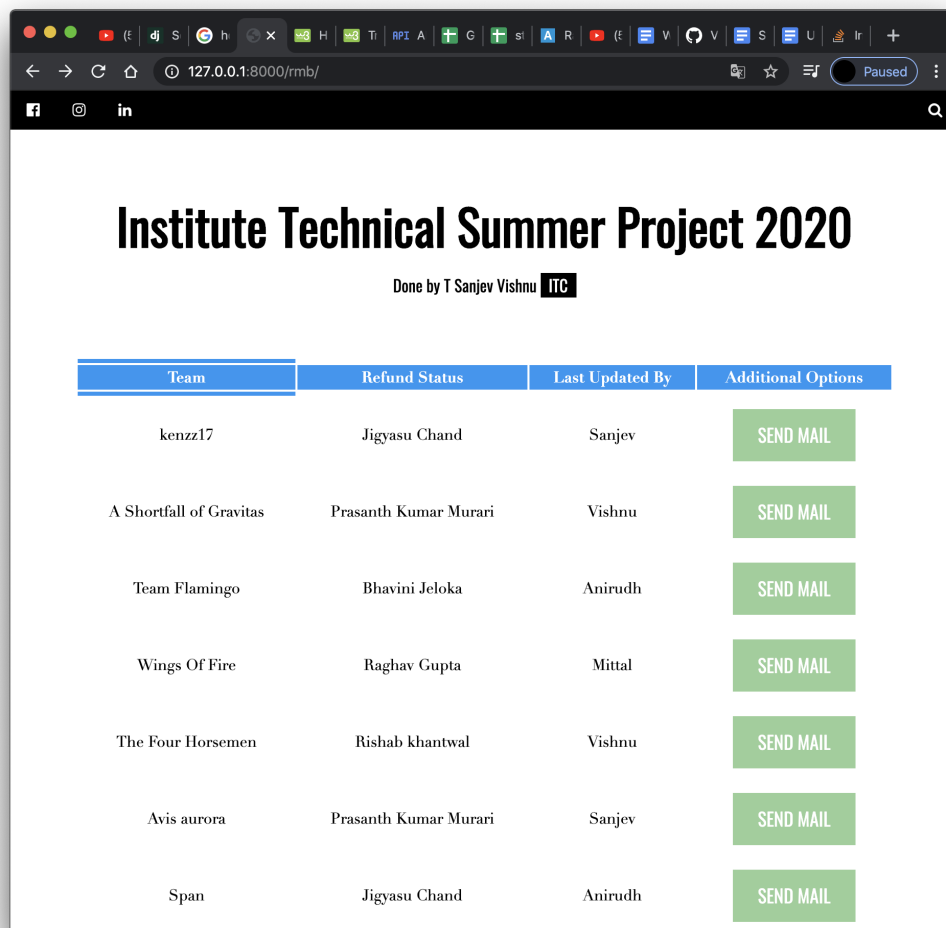You should be able to see numbers printing from 1 to 19 and the following message

```
$ Django version 3.0.5, using settings 'itsp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Open a web browser and go to 127.0.0.1:8000/rmb/
You should be able to see the following page given in the following figure.

There are a list of errors which can come up

2.1 What to do if you get an Error?

If this error pops up, it means the rate at which we read from the database i.e the google sheet is higher than the limit set by Google. So, press **ctrl+c** and run the previous code again.

```
$ gspread.exceptions.APIError: {'code': 429, 'message': "Quota exceeded for
    quota group 'ReadGroup' and limit 'Read requests per user per 100 seconds'
    of service 's
heets.googleapis.com' for consumer 'project_number:355044233060'.", 'status':
    'RESOURCE_EXHAUSTED', 'details': [{'@type':
    'type.googleapis.com/google.rpc.Help', '
links': [{'description': 'Google developer console API key', 'url':
    'https://console.developers.google.com/project/355044233060/apiui/credential'}]}]}
```

If the problem still persists, open the following code by typing this command

```
$ open rmb/views.py
```

Now you have to comment a few lines of code as shown

```python
from django.shortcuts import render
from . models import Status
from . import tsv1
#Comment the following lines
"""
for i in range(1,10):
    print(i)
    new_entry_col =
        Status(tsv1.output_sheet(i,1),tsv1.output_sheet(i,2),tsv1.output_sheet(i,3),tsv1.ou
    new_entry_col.save()

for i in range(10,20):
    print(i)
    new_entry_col =
        Status(tsv1.output_sheet(i,1),tsv1.output_sheet(i,2),tsv1.output_sheet(i,3),tsv1.ou
    new_entry_col.save()
"""
def homepage(request):

    status_lines = Status.objects.all()
    return render(request,'main.html',{ 'status_lines': status_lines })
```
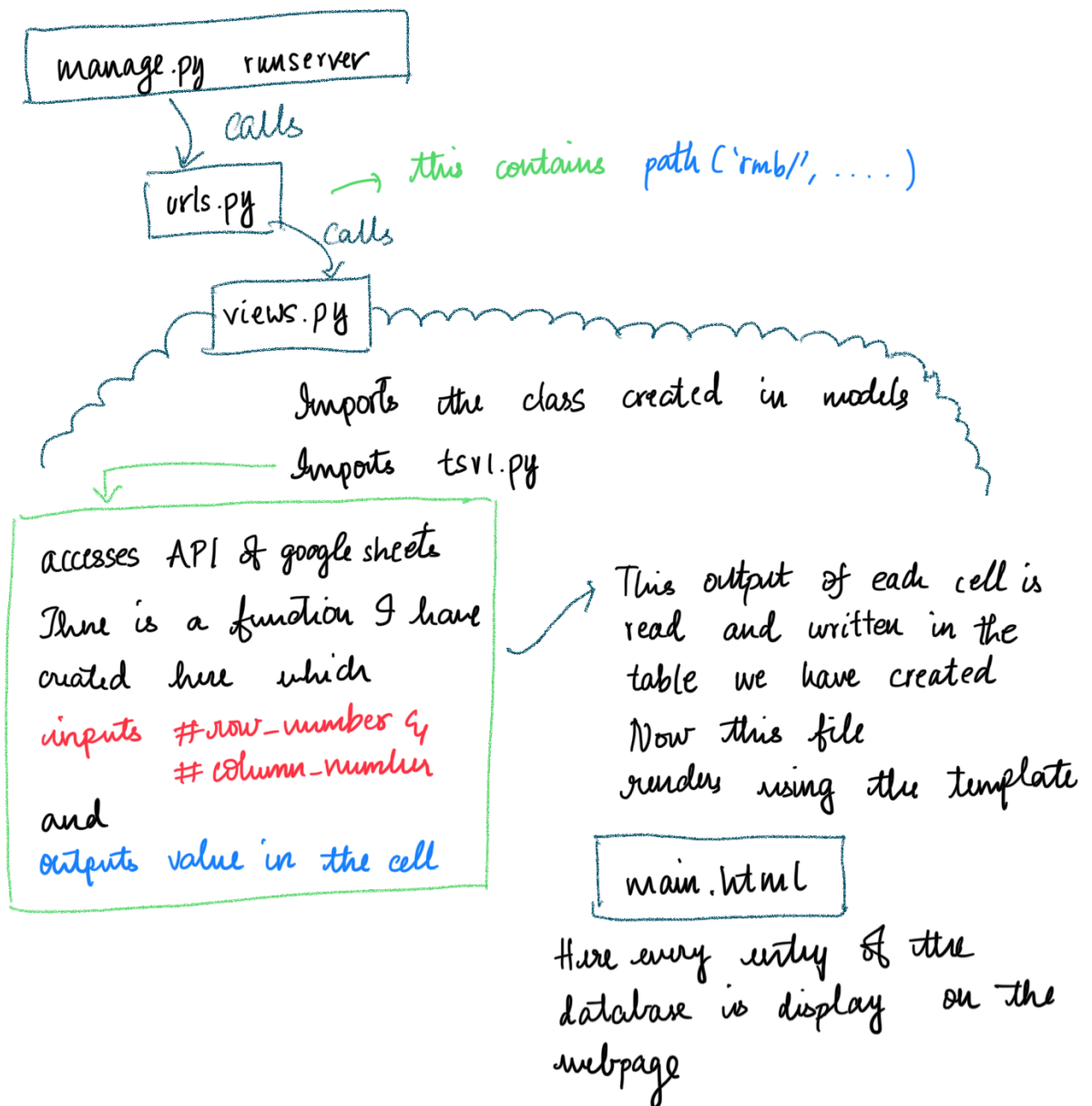
Now, the code should run properly

3 Working of the code

3.1 Tools used

Python3, Django, Google Sheets API, html, css

3.2 Flow Chart

- The database of the model is created by the file models.py in itsp/rmb/

- The name of the class is **Status**

- The fields are title, bill status and last updated by

- The CSS template used here to style some aspects of this webpage is taken from W3schools free templates

manage.py runserver

calls

urls.py → this contains path ('rmb/', ...)

calls

views.py

Imports the class created in models

Imports tsvi.py

accesses API of google sheets
There is a function I have created here which
inputs #row_number &
#column_number
and
outputs value in the cell

This output of each cell is read and written in the table we have created
Now this file renders using the template

main.html

Here every entry of the database is display on the webpage

3.3 Things to tinker

- You can click on the facebook icon and it will re-direct you to tech@iitb fb page

- The google sheet used is https://docs.google.com/spreadsheets/d/16C6znw_BkeHQxKVd8OgAsbEsxxKUD3j_A_cE3YFAYLg/edit?usp=sharing

- You can make changes in the database and run the server again. You will be able to see the changes being reflected on the webpage (**but you can make changes only in the first 19 rows**

- The SEND MAIL button is to send mail to the team's mail id and is not fully developed yet

Since I do not have details of ITSP 2019, I have used the the ITSP 2020 database and instead of the refund status, I have used the mentors' names.

4 Debugging

The above restriction of being able to edit only the first 19 rows is due to the restriction imposed by google API as there is a particular limit on the number of requests accessing API per second. This makes google sheets a non reliable database. So, we need to make our own SQL database on our server instead of relying on third-party sources.