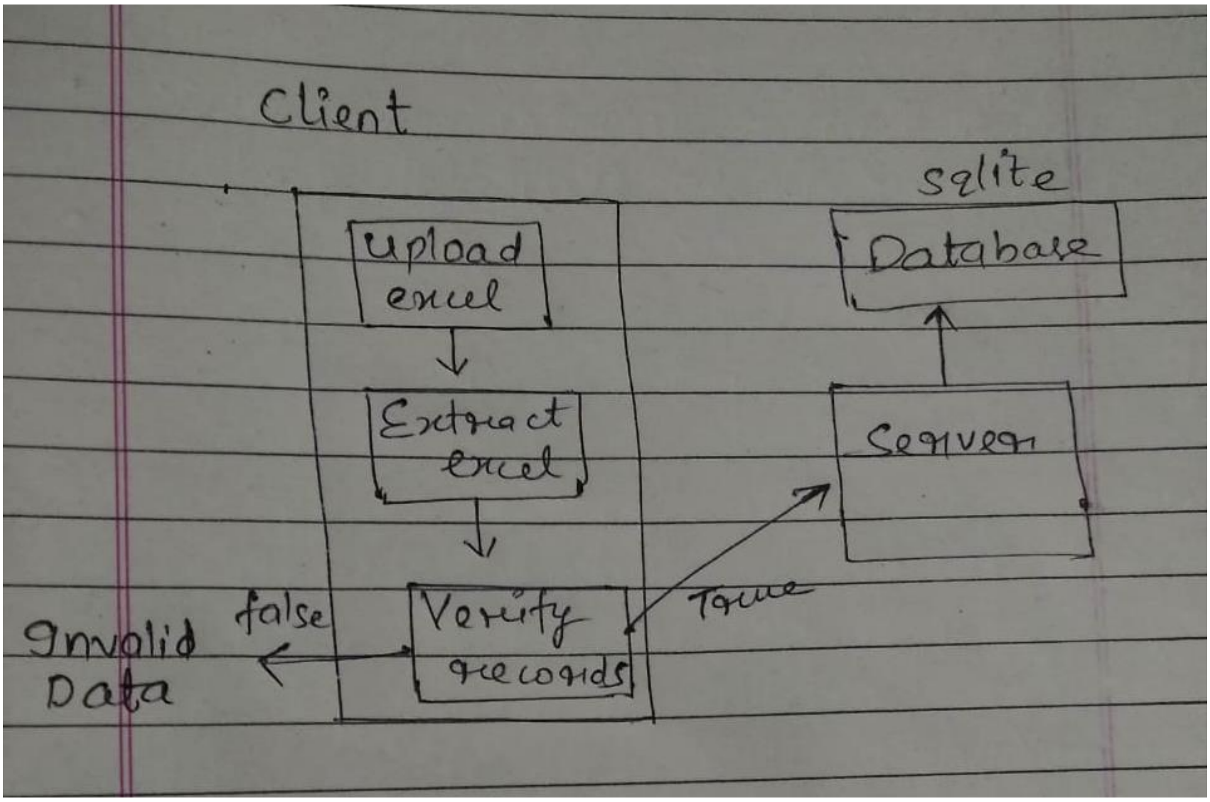
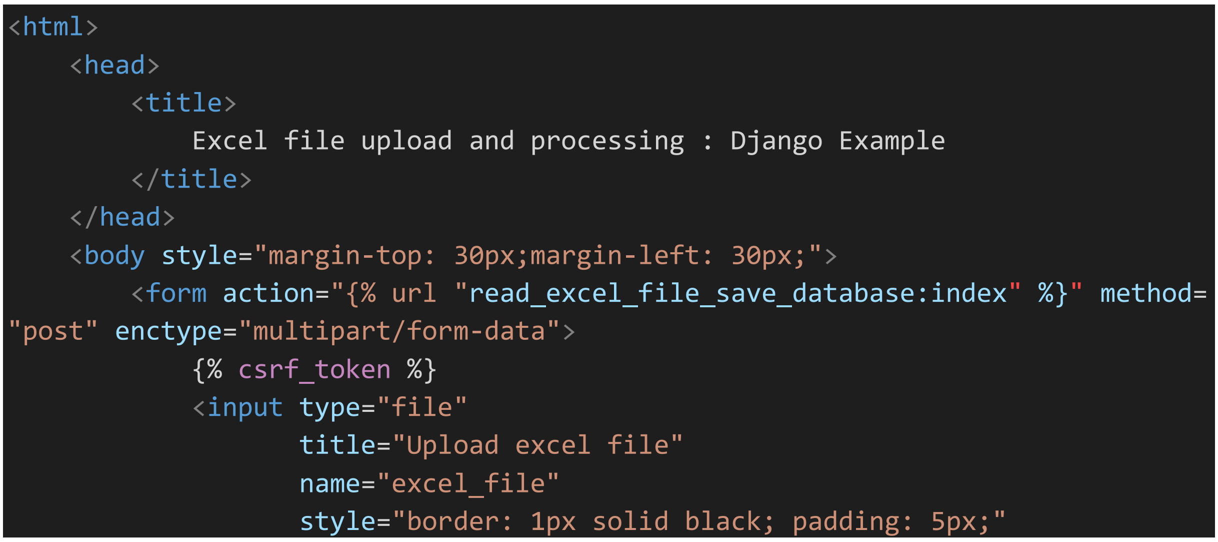
Steps:

1. Make a view where user can upload the xls file.
2. Open the file with openpyxl.load_workbook(filename).
3. Extract, create dict to map the data you want to sync in db.
4. Use the models to add, update or delete the information.



1. Make a view where user can upload the xls file.

Code of Html page to upload xls file :



```
<html>
    <head>
        <title>
            Excel file upload and processing : Django Example
        </title>
    </head>
    <body style="margin-top: 30px;margin-left: 30px;">
        <form action="{% url "read_excel_file_save_database:index" %}" method=
"post" enctype="multipart/form-data">
            {% csrf_token %}
            <input type="file"
                   title="Upload excel file"
                   name="excel_file"
                   style="border: 1px solid black; padding: 5px;"
                   required="required">
            <p>
            <input type="submit"
                   value="Upload"
                   style="border: 1px solid green; padding:5px; border-
radius: 2px; cursor: pointer;">
        </form>

        <p></p>
        <hr>

        {% for row in excel_data %}
            {% for cell in row %}
                {{ cell }}&nbsp;&nbsp;
            {% endfor %}
            <br>
        {% endfor %}
    </body>
</html>
```

2. **Database Model in Django to upload data:**

```
from django.db import models

# Create your models here.
class student(models.Model):
        reg = models.CharField(max_length=20)
        ses = models.CharField(max_length=20)
        sem= models.DecimalField(max_digits=12, decimal_places=2)
        sem_type = models.CharField(max_length=20)
        prog = models.CharField(max_length=20)
        bran = models.CharField(max_length=20)
        spi = models.DecimalField(max_digits=12, decimal_places=2)
        p_cpi = models.DecimalField(max_digits=12, decimal_places=2)
        cpi = models.DecimalField(max_digits=12, decimal_places=2)
        result = models.CharField(max_length=20)
```

3. **Open excel file and extract data:**

```
from django.shortcuts import render
from django.http import HttpResponse
import re
# Create your views here.

import openpyxl
```

```
from read_excel_file_save_database.models import student

def index(request):
    if "GET" == request.method:
        return render(request, 'index.html', {})
    else:
        excel_file = request.FILES["excel_file"]

        # you may put validations here to check extension or file size

        wb = openpyxl.load_workbook(excel_file)
```

4. **Check whether registration no. of of the form registration_year - CS - enrollment no:**

```
row_no=0
        for row in worksheet.iter_rows():
            row_data = list()
            for cell in row:
                if cell.value!=None :
                    row_data.append(str(cell.value))
            if not row_data or row_no==0:
                excel_data.append(row_data)
                row_no+=1
                continue
            st = student()
            st.reg = row_data[0]
            st.ses = row_data[1]
            st.sem= row_data[2]
            st.sem_type = row_data[3]
            st.prog = row_data[4]
            st.bran = row_data[5]
            st.spi = row_data[6]
            st.p_cpi = row_data[7]
            st.cpi = row_data[8]
            st.result = row_data[9]
            n = 4
            myset = {"DN","BM","BT","CL","EN","GE","ST","CM","CC","CS","EE","PE","EM","FE","GI","IS","MT","PR","TR","VL","PS","PD","TH","SP","SW"}
            split = [st.reg[i:i+n] for i in range(0, len(st.reg), n)]
            match = re.match(r'.*([1-3][0-9]{3})', split[0])
            if match is not None :
                if split[1] == "PTSW" :
                    st.save()
                    excel_data.append(row_data)
                else :
                    n1=2
                    str1=split[1]
                    split1 = [str1[i:i+n1] for i in range(0, len(str1), n1)]
                    if split1[0] in myset :
                        st.save()
                        excel_data.append(row_data)
                    else :
                        row_data1 = list()
                        row_data1.append(str("Invalid Data of row no. "+ str(row_no)))
                        excel_data.append(row_data1)
            else :
                row_data1 = list()
                row_data1.append(str("Invalid Data of row no. "+ str(row_no)))
                excel_data.append(row_data1)
            row_no+=1

        return render(request, 'index.html', {"excel_data":excel_data})
```

if registration no. is of given form then we save the data of that row in database and print the same data on screen and if the registration no. is not of the given form then we print Invalid Data of row no.