

FARM IN ARMS

A

Project Report

*Submitted in partial fulfilment of the Requirements for the award of
the Degree of*

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

A. MONICA (1602-18-737-084)

B. SAI MANISHA (1602-18-737-097)

M. SANNIHITHA (1602-18-737-104)

Under the guidance of

GAYATHRI

ASSISTANT PROFESSOR



Department of Information Technology

Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University)

IbrahimBagh , Hyderabad-31

2019-2020

**Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Hyderabad-500 031**

Department of Information Technology

DECLARATION BY

THE CANDIDATE



We **A. MONICA, B. SAI MANISHA, M. SANNIHITHA** bearing hall ticket number **1602-18-737-084, 1602-18-737-097, 1602-18-737-104** hereby declare that the project report entitled “**FARM IN ARMS**” under the guidance of **GAYATHRI**, Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering in Information Technology**.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

A. MONICA (1602-18-737-084)

B. SAI MANISHA (1602-18-737-097)

M. SANNIHITHA (1602-18-737-104)

Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Hyderabad-500 031

Department of Information Technology



BONAFIDE CERTIFICATE

This is to certify that the project entitled “**FARM IN ARMS**” being submitted by **A. MONICA, B. SAI MANISHA, M. SANNIHITHA** bearing hall ticket number **1602-18-737-084, 1602-18-737-097, 1602-18-737-104**, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Engineering in Information Technology** is a record of bonafide work carried out by him/her under my guidance.

GAYATHRI

Assistant Professor

Dr. K. RAM MOHAN RAO

HOD, IT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We would like to take the opportunity to express our humble gratitude to **GAYATHRI** under whom we executed this project.

We would also use this opportunity to thank our senior **LAHARI POKALA (1602-17-737- 014)**. We are grateful to her guidance, and constructive suggestions that helped us in the preparation of this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks. We would like to thank all faculty members and staff of the **Department of Information Technology** for their generous help in various ways for the completion of this project.

Finally, yet importantly, we would like to express our heartfelt thanks to our HOD **Dr.K. Ram Mohan Rao** and classmates for their help and wishes for the successful completion of this project.

ABSTRACT

In agriculture, activities such as selecting the right crop which is suitable to that particular weather conditions, precautions to be taken while growing the crop, selective spraying of agrochemicals are essential to maintaining high productivity and quality of yield. Our system provides details to farmers on weather patterns that could impact the crop, possible pest attacks, weather updates, steps for producing high yield, regional terrain. Our website serves as a crop guidance system to help and assist farmer in producing better crop.

List of Contents –

1) Introduction	7
2) Related work	7
3) Proposed System	8
4) Proposed work	8-25
a) Use cases	8
b) Screen shots of UI	9-10
c) Technology and Architecture used	10-11
d) Design	12-14
i) UML static diagrams	12-13
(1) Class diagram	
(2) Activity diagram	
(3) Sequence diagram	
ii) UML run time diagrams	14
(1) Flowchart diagram	
e) Implementation	16-25
f) Testing	25
g) Github Link	26
5) Results	26-32
6) Discussion and Future Work.....	33
7) References	33

Introduction

This project is built using Django framework. Django is an open source web application framework which is written in python. This crop guidance system built using Django has three major components each of which has different functionality but similar architecture. The technique and process which is applied in this project can be used in the future for the extension of the project and other complex database-driven websites.

Farm in arms – a project developed using Django is useful for farmers in producing better crop. This project is an interactive and informational web-based platform that helps farmers by suggesting suitable crop that can be grown in that particular area, providing guidance on the cultivation of the crop, suggesting best agrochemicals and providing weather updates and weather alerts daily. Our website serves as a crop guidance system to help and assist farmer in producing better crop.

Related work

There are many agriculture related apps in the market which emphasize on crop guidance. Some of them mainly focus on market place and risk management while others give a description of cultivation of crops.

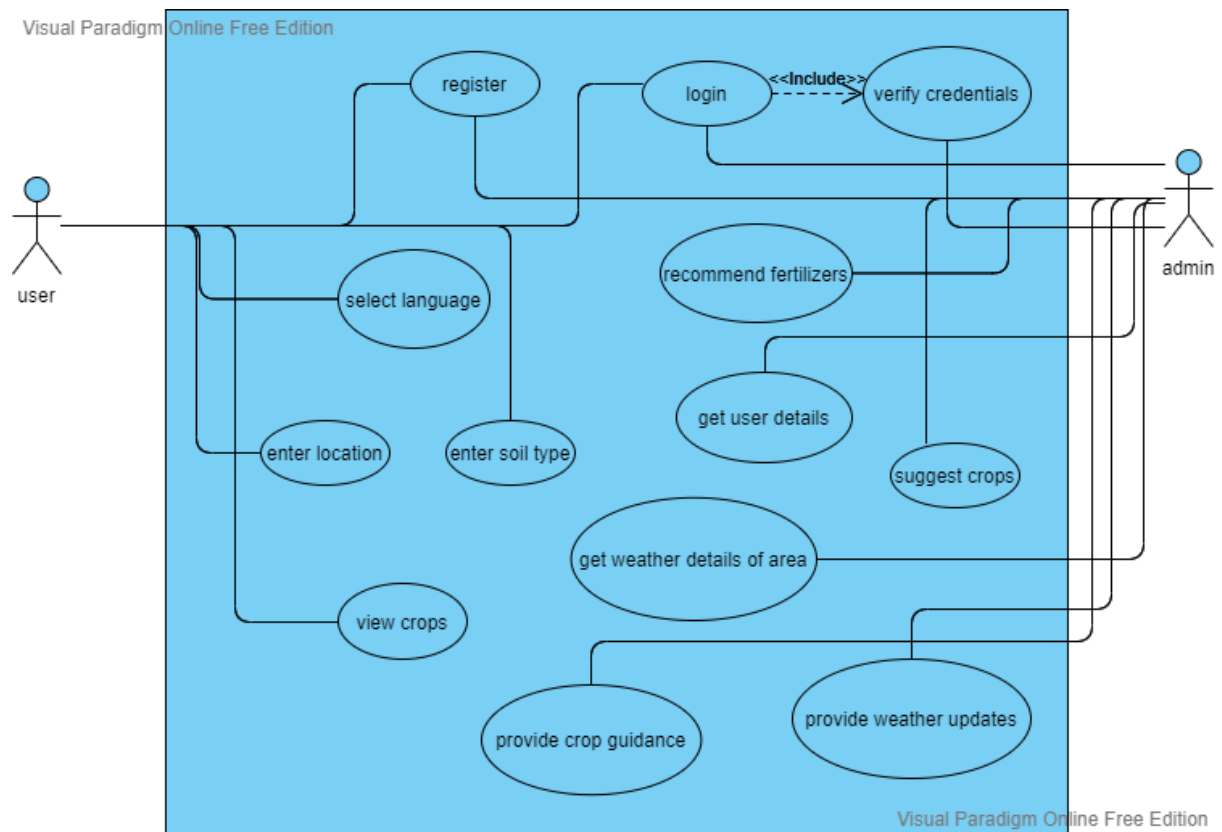
The aspect that is different in our system is that we also analyse the weather pattern of a particular area and suggest the crop to the farmers which is suitable to grow in those climatic conditions. We also focus on providing ultimate care for crop management from cultivating to the harvest.

Proposed System

The proposed system is fully interactive and efficient, which can be used and understood by everyone. Farmers should login/register and select their region and language. Our system analyses the weather conditions of that area and suggests the crops which are suitable to grow in those climatic conditions. Based on the crop selection farmer will be guided in cultivating the crop from begin to end – proper steps to be taken while cultivation, determining crop water need, crop ailments and best suitable fertilizers and pesticides. Weather updates and weather alerts are also provided on a daily basis.

Proposed Work

a. Use case diagram



b. Screenshots of UI:

LOG IN FORM

127.0.0.1:8000/accounts/login/

Translate to:

Select Language

Powered by Google Translate

Back

Log In

Mobile Number

Password

☒ Keep me signed in

Submit

Not a member? [Register](#)


127.0.0.1:8000/accounts/register/

Translate to:

Select Language

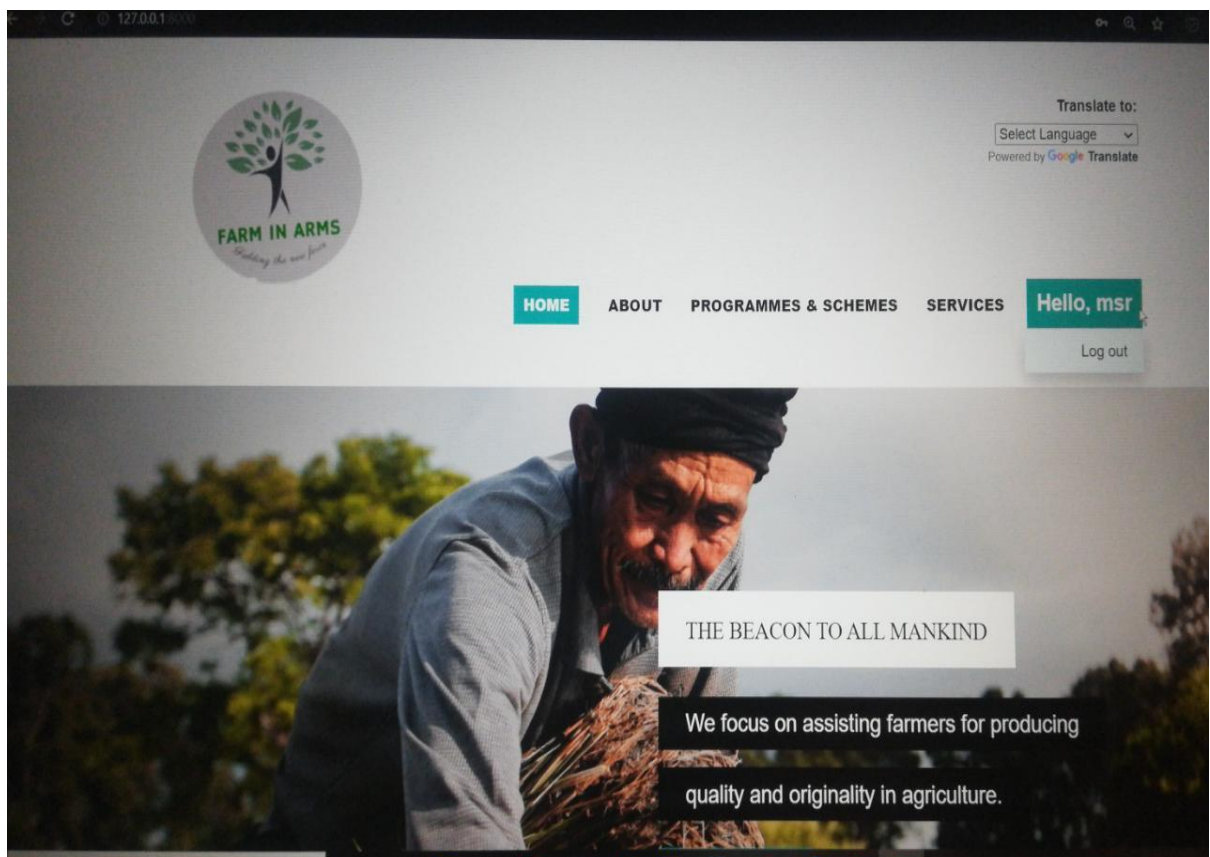
Powered by Google Translate

Back



SIGN UP

Already Have an Account? [Log In](#)



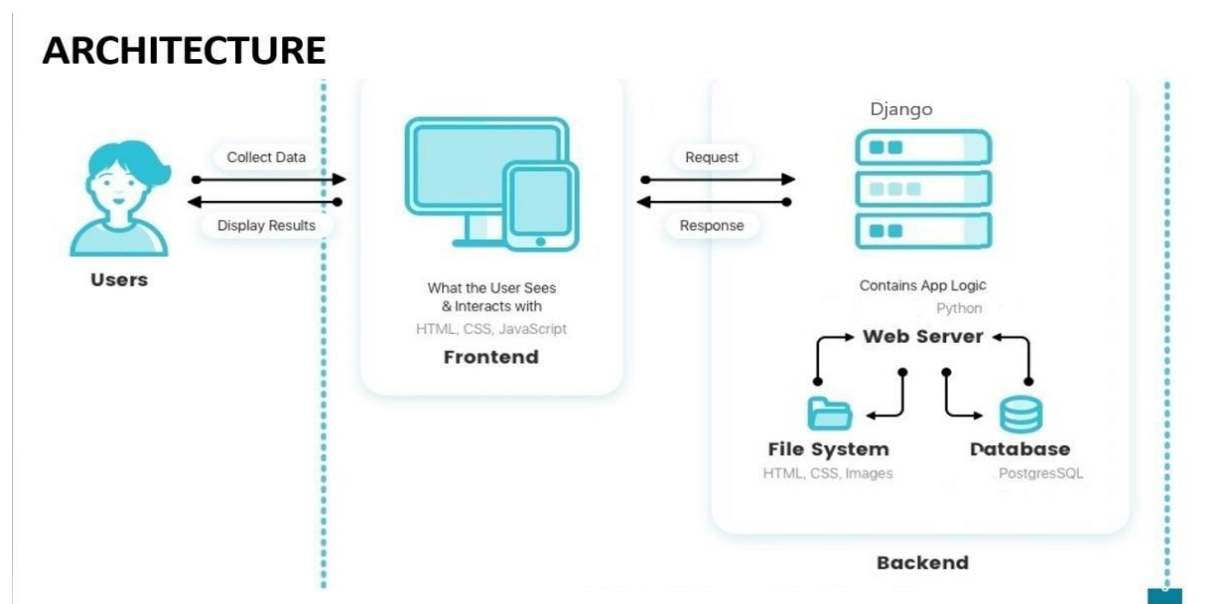
c. Technology and Architecture used:

Various technologies are used for building the project

For developing UI:

- HTML
- CSS
- Javascript
- Bootstrap

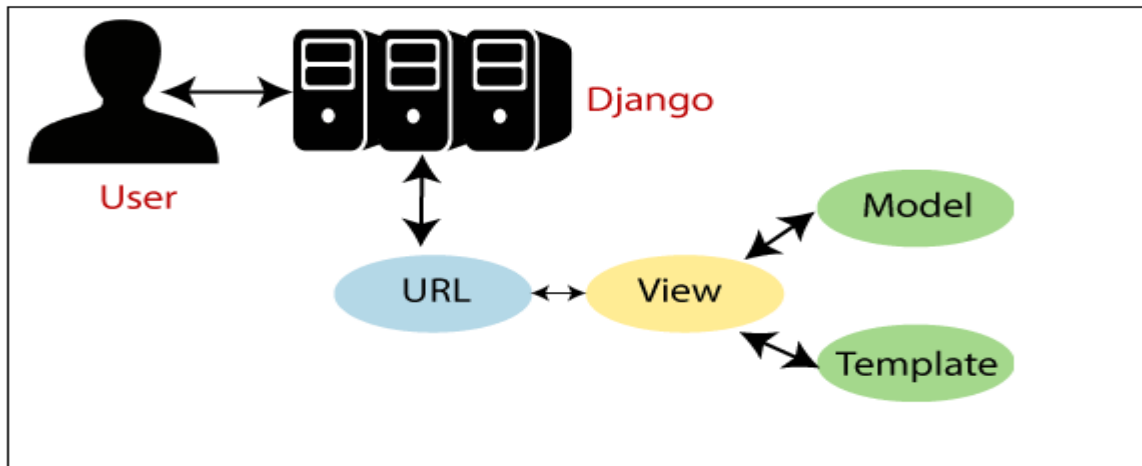
At the backend, postgresql database was used and Django framework is used to develop the project. There are 3 major modules- accounts, weather and crop modules. First super user has to be created in order to gain the admin rights. Every person must have phone number and password to access. MVT framework in Django enables us to connect to the database and retrieve values from it. It also consists inbuilt middleware such as csrf token to protect against CSRF attacks by generating a csrf token in the server, send it to client and mandating the client to send it back in the request header.



Django MVT:

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data. The Template is a

presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template. Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the framework itself. There is no separate controller and complete application is based on Model View and Template. That's why it is called MVT application. See the following graph that shows the MVT based control flow.

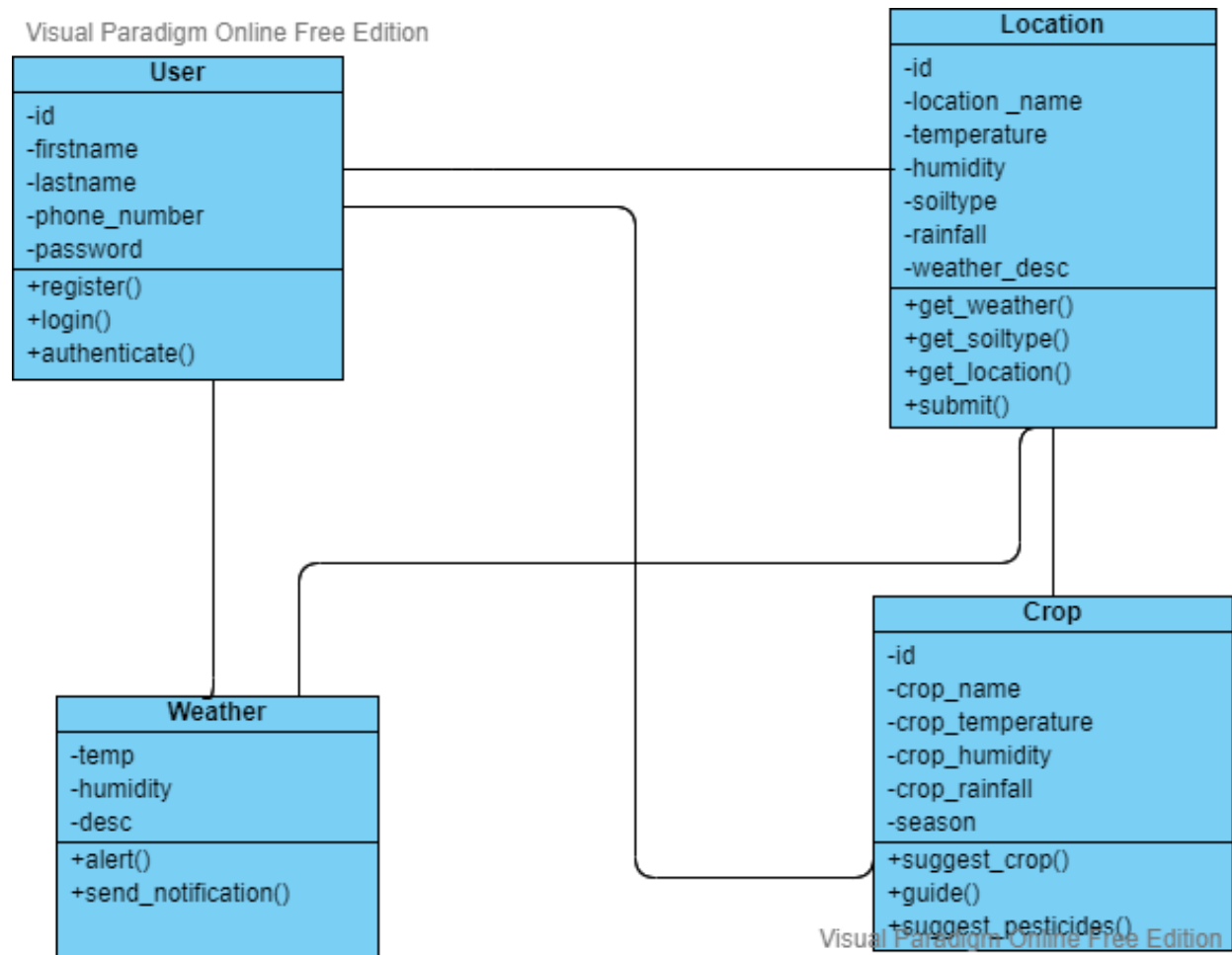


d. Design:

i. UML Static diagrams:

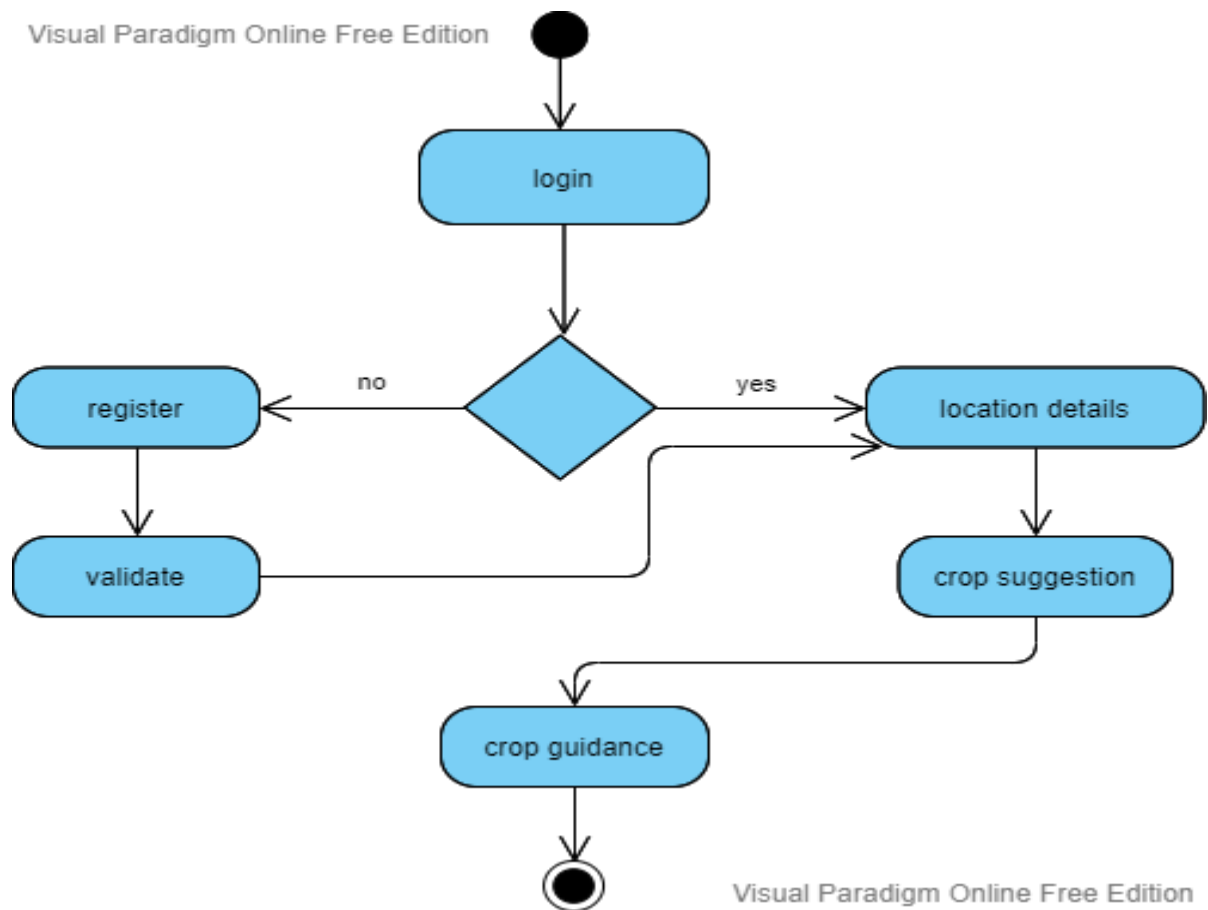
class diagram-

The below figure depicts the class diagram of a system.

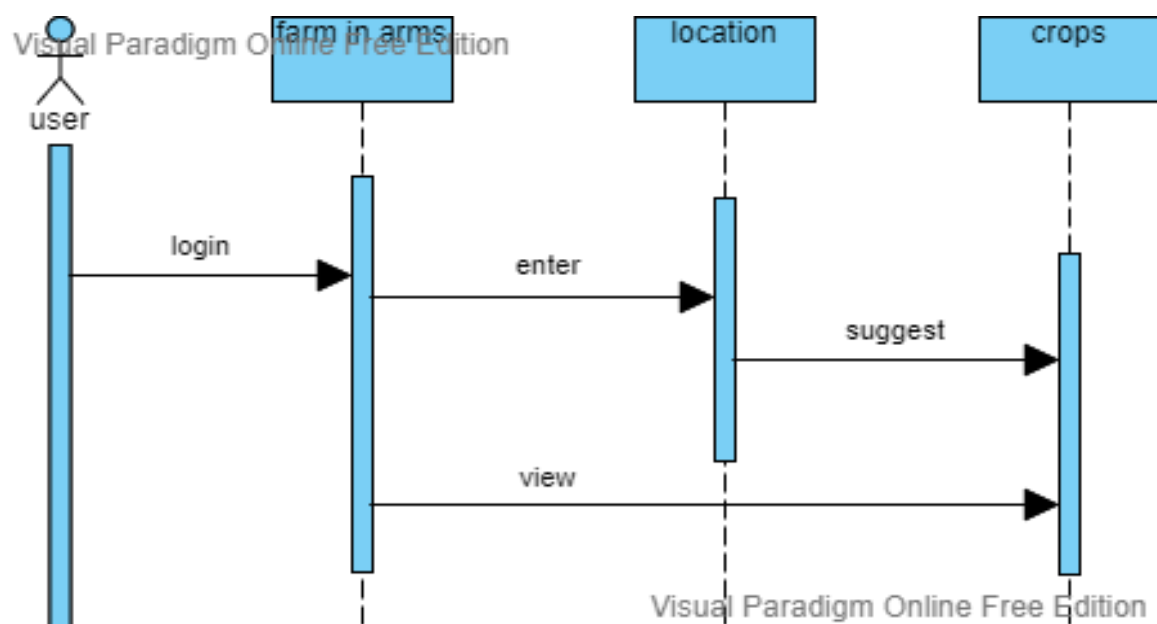


Activity Diagram-

The below figure depicts the activity diagram of a system.

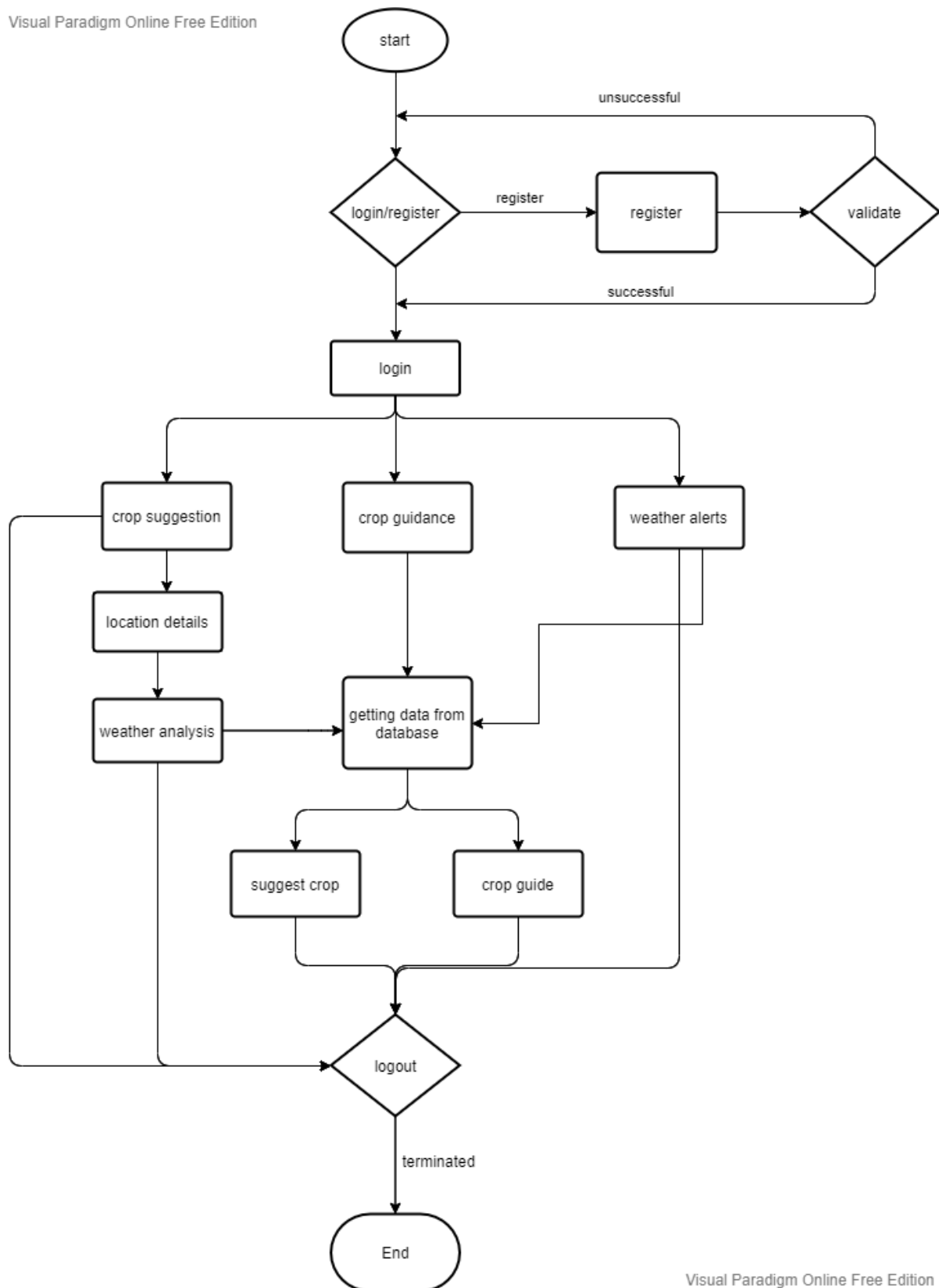


Sequence diagram-



Flowchart Diagram-

Visual Paradigm Online Free Edition



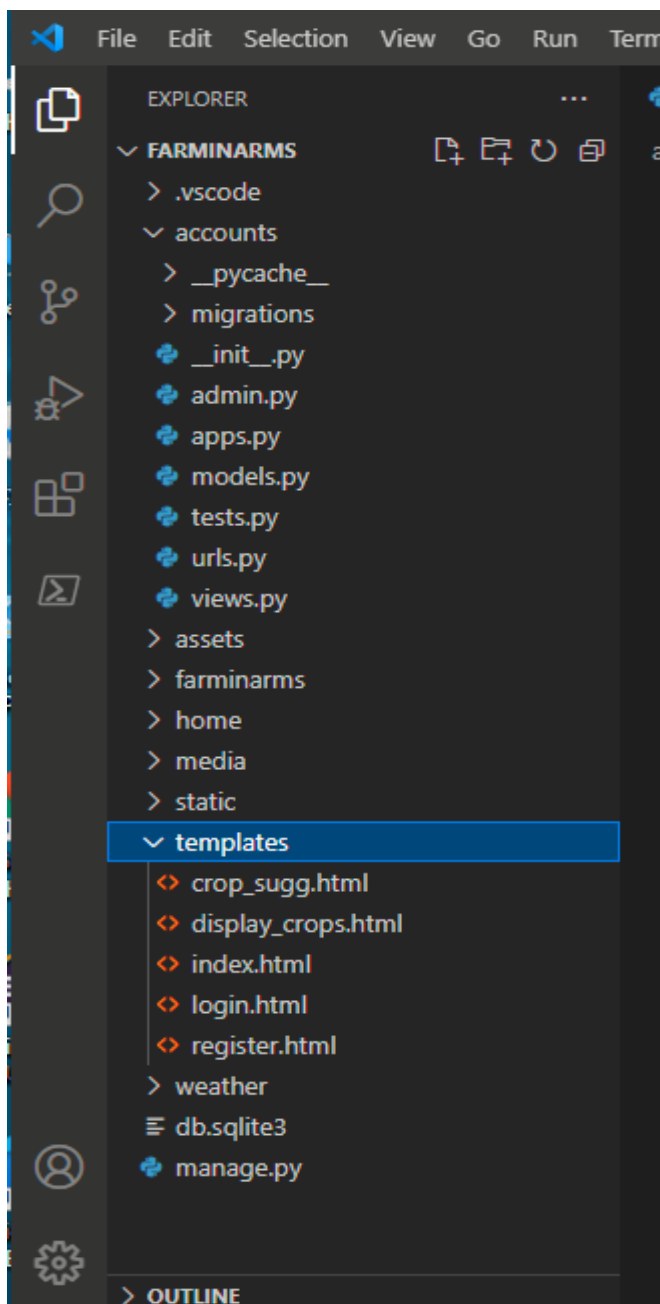
Visual Paradigm Online Free Edition

e. Implementation:

Main Modules

Accounts module:

This module is responsible for managing user and admin accounts. All the data related to the customer accounts is stored using Django models of Accounts module. Whenever a customer tries to login to his account the validation is done internally where the model component fetches the data from the database and validates further. This can be accessed by the admin using the admin dashboard.



Model

```
from django.db import models
from django.contrib.auth.models import AbstractUser, BaseUserManager
from django.core.validators import RegexValidator
from django.utils.translation import ugettext_lazy as _
# Create your models here.

class UserManager(BaseUserManager):
    """Define a model manager for User model with no username field."""

    use_in_migrations = True

    def _create_user(self, phn, password, **extra_fields):
        """Create and save a User with the given phone and password."""
        if not phn:
            raise ValueError('The given phone must be set')

        user = self.model(phn=phn, **extra_fields)
        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_user(self, phn, password=None, **extra_fields):
        """Create and save a regular User with the given phone and password."""
        extra_fields.setdefault('is_staff', False)
        extra_fields.setdefault('is_superuser', False)
        return self._create_user(phn, password, **extra_fields)

    def create_superuser(self, phn, password, **extra_fields):
        """Create and save a SuperUser with the given phone and password."""
        extra_fields.setdefault('is_staff', True)
        extra_fields.setdefault('is_superuser', True)

        if extra_fields.get('is_staff') is not True:
            raise ValueError('Superuser must have is_staff=True.')
        if extra_fields.get('is_superuser') is not True:
            raise ValueError('Superuser must have is_superuser=True.')

        return self._create_user(phn, password, **extra_fields)

class User(AbstractUser):
    """User model."""

    username = None
    email = models.EmailField(blank=True, null=True)
```

```

phone_regex = RegexValidator(regex=r'^(\0|91|\+91)?-
?[789]\d{9}$', message="Phone number must be entered in the format: '+919999999
9999'. Up to 10 digits allowed.")
phn = models.CharField(_('phone number'), validators=[phone_regex], max_le
ngth=17, unique=True) # validators should be a list

USERNAME_FIELD = 'phn'
REQUIRED_FIELDS = []

objects = UserManager()

```

View

```

from django.core.checks import messages
from django.shortcuts import render, redirect
from django.contrib.auth.models import auth
from .models import User
# Create your views here.
def register(request):
    if request.method == 'POST':
        first_name = request.POST['fname']
        last_name = request.POST['lname']
        password1 = request.POST['upass1']
        password2 = request.POST['upass2']
        phn = request.POST['phno']
        if(password1 == password2):
            #if User.objects.filter(phn=phn).exists:
                # messages.Info('request','This Phone number is already registe
red')
                #return redirect('register')
            user = User.objects.create_user(first_name=first_name,last_name=la
st_name,password=password1,phn=phn)
            user.save()
            return redirect('/accounts/login/')
        else:
            print('passworn not matching.')
            return redirect('/')
    else:
        return render(request,'register.html')

def login(request):
    if request.method == 'POST':
        phn = request.POST['phn']
        password = request.POST['upass']

```

```

        user = auth.authenticate(phn=phn,password=password)

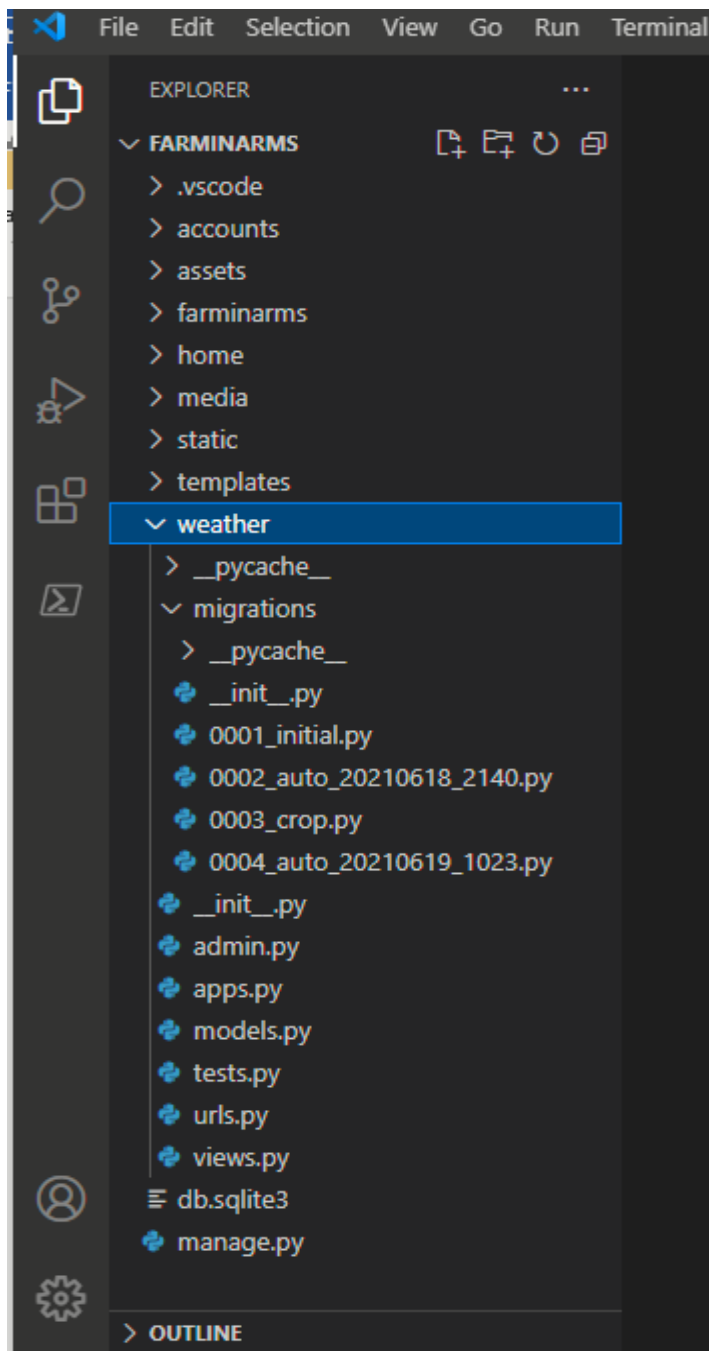
        if user is not None:
            auth.login(request,user)
            return redirect('/')
        else:
            messages.Info(request,'Invalid Credentials!')
            return redirect('login')
    else:
        return render(request,'login.html')

def logout(request):
    auth.logout(request)
    return redirect('/')

```

Weather and Crops module –

This module is responsible for taking the location and soil details from the user and storing them in the database. Through openweathermaps api , it analyses the weather conditions of the area such as temperature, humidity, rainfall,description, etc... and stores them in the database. Some of the details of major crops cultivated in Telangana are also stored in the database and based on the weather values of the location obtained from the openweathermaps, it is mapped with those present in the crop table. Through nested queries and joins, we extract specific crops suitable to grow in that location and display them.



Model

```
from django.db import models
from django.db.models.deletion import DO_NOTHING
from accounts.models import User

# Create your models here.
class location(models.Model):
    name = models.CharField(max_length=20, null=False)
    temp = models.FloatField(null=True)
```

```

soilt = models.CharField(max_length=20,null=True)
humd = models.FloatField(null=True)
desc = models.CharField(max_length=50,null=True)
precip = models.FloatField(null=True)
lat = models.FloatField(null=True)
lon = models.FloatField(null=True)

cust = models.ForeignKey(User,on_delete=models.DO_NOTHING,null=True)

def _str_(self):
    return self.name

class crop(models.Model):
    cname=models.CharField(max_length=30,null=False)
    cimg=models.ImageField(upload_to='pics')
    cmintemp=models.FloatField(null=True)
    cmaxtemp=models.FloatField(null=True)
    cminhum=models.FloatField(null=True)
    cmaxhum=models.FloatField(null=True)
    cminprecip=models.FloatField(null=True)
    cmaxprecip=models.FloatField(null=True)
    csoiltype=models.CharField(max_length=30,null=True)

    croploc=models.ForeignKey(location,on_delete=DO_NOTHING,null=True,blank=True)

    def __str__(self):
        return self.cname

```

View

```

from django.shortcuts import redirect, render
import requests
from .models import location,crop
from accounts.models import User

# Create your views here.
def det(request):
    if request.method == 'POST':
        cname = request.POST['cname']
        soilt = request.POST['soiltype']
        api_key = "8d7fc4b01819676b788ed6f1048470bd"
        base_url = "http://api.openweathermap.org/data/2.5/weather?"
        curl=base_url + 'q='+ cname + '&appid=' + api_key

        response = requests.get(curl)
        x = response.json()

```

```

        if x["cod"] != "404":

            y = x["main"]
            temp = y["temp"]-273      #Degree Centigrade
            humd = y["humidity"]      #percentage

            z = x["weather"]
            desc = z[0]["description"]

            obj = location(name=cname,soilt=soilt,temp=temp,humd=humd,desc=desc,cu
st=request.user)
            obj.save()
            return render(request,'display_crops.html')
        else:
            return render(request,'crop_sugg.html')

def cropdet(request):
    dest=crop.objects.all()
    return render(request,"display_crops.html",{ 'dest':dest})

```

Database Connection

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'farminarms',
        'USER': 'postgres',
        'PASSWORD': 'manisha@0514',
        'HOST': 'localhost'
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },

```

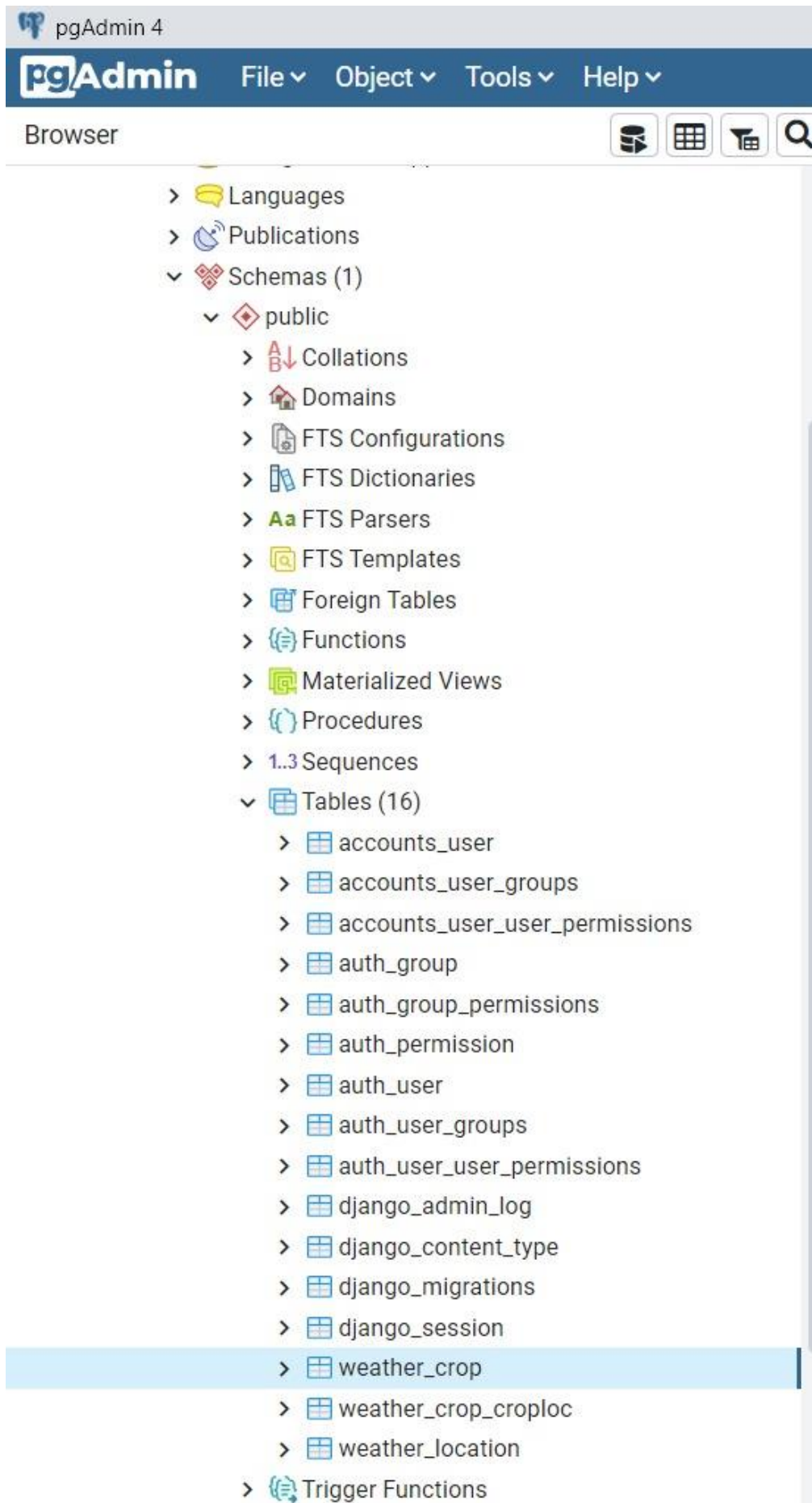
```

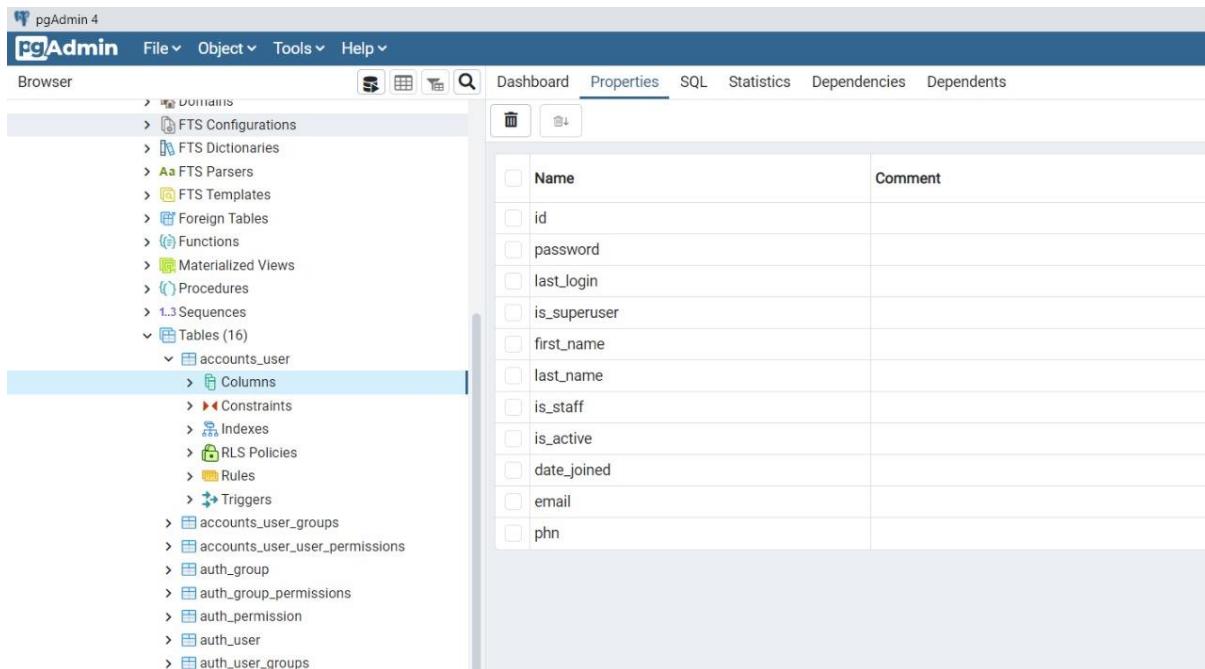
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValida
tor',
    },
]

```

Tables in database – Postgresql







f. Testing

Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses. Following tests are made so that the application runs

Registration:

- Phone number – is necessary, should match format and should not have duplicates in database.
- Password – is necessary, must not be visible to user and should not have duplicates in database.
- First name- is necessary and should have only alphabets in database.
- Email- must match the format.

Login:

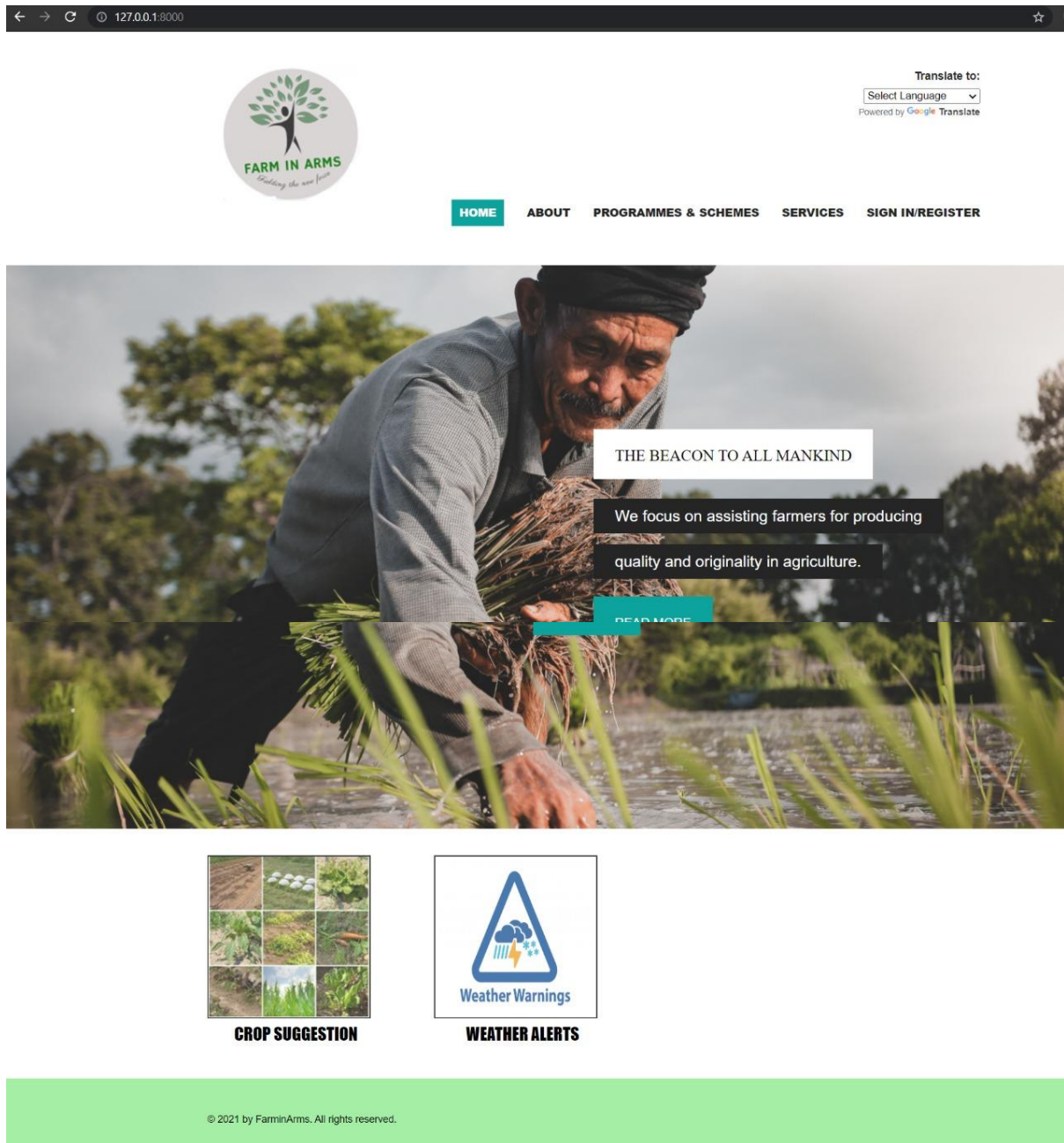
- Phone number and Password- should match with that of in database else could not log in.

g. Github Link

<https://github.com/SannihithaReddy/farminarms/tree/master>

Results

Home Page



Registration form


← → ↻ 127.0.0.1:8000/accounts/register/ ☆

Translate to:

Select Language ▼

Powered by Google Translate

Back



SIGN UP

Already Have an Account? [Log In](#)

Login Form

LOG IN FORM × +


← → ↻ 127.0.0.1:8000/accounts/login/

Translate to:

Select Language ▼

Powered by Google Translate

Back

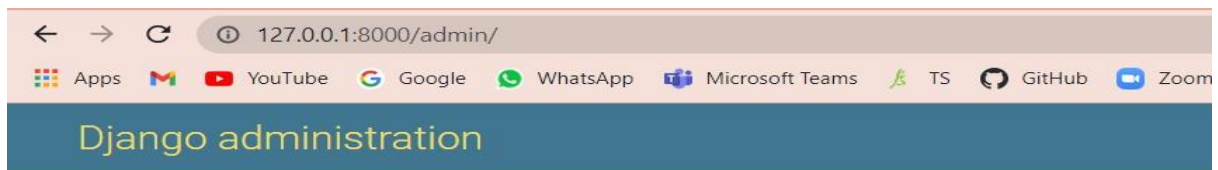


Log In

☒ Keep me signed in

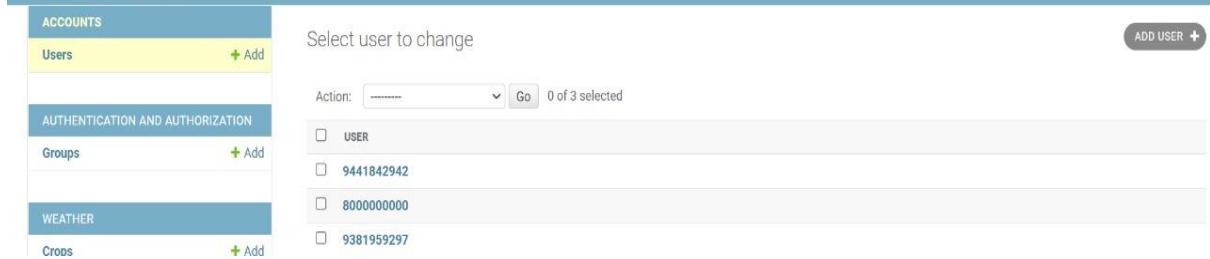
Not a member? [Register](#)

Admin page



Site administration

ACCOUNTS		
Users	+ Add	Change
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
WEATHER		
Crops	+ Add	Change
Locations	+ Add	Change



Django administration

Home > Weather > Locations

ACCOUNTS

Users [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

WEATHER

Crops [+ Add](#)

Locations [+ Add](#)

Select location to change

Action: 0 of 4 selected

☐ LOCATION

☐ habsiguda

☐ guntur

☐ hanamkonda

☐ hyderabad

127.0.0.1:8000/admin/weather/crop/2/change/

Django administration

Home > Weather > Crops > cotton

WELCOME MSR. VIEW SITE / CHANGE PASSWORD / LOG OUT

ACCOUNTS

Users [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

WEATHER

Crops [+ Add](#)

Locations [+ Add](#)

Change crop

cotton

HISTORY

Cname:

Cimg: Currently: pics/cotton.PNG
Change: No file chosen

Cmintemp:

Cmaxtemp:

Cminhum:

Cmaxhum:

Cminprecip:

Cmaxprecip:

Django administration

WELCOME MSR. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Weather > Locations > guntur

ACCOUNTS

Users [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

WEATHER

Crops [+ Add](#)

Locations [+ Add](#)

Change location

guntur

HISTORY

Name:

Temp:

Soilt:

Humd:

Desc:



Translate to:

 Powered by Google Translate

HOME ABOUT PROGRAMMES & SCHEMES SERVICES SIGN IN/REGISTER



Agriculture is the backbone of the Indian Economy"- said Mahatma Gandhi six decades ago.

Even today, the situation is still the same, with almost the entire economy being sustained by agriculture, which is the mainstay of the villages. It contributes 16% of the overall GDP and accounts for employment of approximately 52% of the Indian population. Rapid growth in agriculture is essential not only for self-reliance but also to earn valuable foreign exchange.

Indian farmers are second to none in production and productivity despite of the fact that millions are marginal and small farmers. They adopt improved agriculture technology as efficiently as farmers in developed countries. It is felt that with provision of timely and adequate inputs such as fertilizers, seeds, pesticides and by making available affordable agricultural credit /crop insurance, Indian farmers are going to ensure food and nutritional security to the Nation. It is envisaged to make available relevant information and services to the farming community and private sector through the use of information and communication technologies, to supplement the existing delivery channels provided for by the department. Farmers' Portal is an endeavour in this direction to create one stop shop for meeting all informational needs relating to Agriculture sectors production of a farmer. With this Farmer will not be required to sift through maze of websites created for specific purposes.

Once in the Farmers' Portal, a farmer will be able to get all relevant information on specific subjects around his village/block /district or state. This information will be delivered in the form of text, SMS in the language he or she understands. These levels can be easily reached through the Map of Telangana placed on the Home page. Farmers will also be able to ask specific queries as well as give valuable feedback through the Feedback module

Crop suggestion and guidance



Translate to:

 Powered by Google Translate

HOME ABOUT PROGRAMMES & SCHEMES SERVICES Hello, msr

Town/Village Name

ZIP code

Select Soil Type:

Red soil

submit



[Home](#)

Suggested Crops



maize



cotton

Crop Guidance



Translate to:

Powered by [Google Translate](#)

[HOME](#) [ABOUT](#) [PROGRAMMES & SCHEMES](#) [SERVICES](#) [Hello, msr](#)





Translate to:

Select Language
Powered by Google Translate

Management

[Land preparation](#)
[Water Management](#)
[Pest Management](#)

Paddy



Land Preparation

Equipment Options And Use

Primary tillage options – Ploughs (Mould board, Disc, Tine, Offset)



Management

[Land preparation](#)
[Water Management](#)
[Pest Management](#)

Water Management

Principles

Excess / limited / no water leads to reduction in yield.
Every crop plant tries to survive under extreme drought conditions but never survives for atleast 1 – 2 days under excess moisture situation.
Rice a semi – aquatic plant requires near submergence.
Submergence helps in - suppressing weed growth more availability of certain nutrients.
Daily consumptive use of rice is 6 – 10mm.
Total water requirement of rice is 1240mm.
5000 litres of water required to produce 1kg of rice.
Rice accounted for 33% of gross irrigated area.
Rice consuming 66% of total available irrigation water.

Pest Management

Losses caused due to pests in Rice

- Rice is essentially a crop of warm, humid environments conducive to the survival and Proliferation of insects.
- More than 70 species were recorded as pests of rice and about 20 have major significance.
- Together, they infest all parts of the plant at all growth stages.
- The insects act as vectors of virus diseases, and are a major factor responsible for low rice yields particularly in Tropical Asia, the world's rice bowl.
- The insect problem is accentuated in multi cropping Or dormancy but occurs throughout the year in over lapping generations.
- The yield losses vary from 20 to 50 per cent due to the damage caused by various insect Pests.

Control Measures

- Control measures are mainly Cultural, Mechanical, Biological & Chemical.
- Among the different methods the farmer is inclined more for chemical method of control as this method gives quick results.
- No single method is adequate to suppress pest population.
- Integrated pest management depending on the need, availability and feasibility of implementation should be adopted.

Advantages of Chemical Control

Weather Updates



Translate to:

Powered by Google Translate

[HOME](#) [ABOUT](#) [PROGRAMMES & SCHEMES](#) [SERVICES](#) [Hello, msr](#)

Enter ZIP code of the area to get the weather details.

Translate to:

Powered by Google Translate

[HOME](#) [ABOUT](#) [PROGRAMMES & SCHEMES](#) [SERVICES](#) [Hello, msr](#)

Enter ZIP code of the area to get the weather details.

Temperature (in Degree Centigrade) = 27.290000000000002
atmospheric pressure (in hPa unit) = 1006
humidity (in percentage) = 57
Weather Description : [overcast clouds](#)

Discussion and future work

Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements with web can be made to the application, so that the web site functions very attractively and more useful than the present one.

- Currently, it is only limited to Telangana. In the future, we can extend it at a national level including all the major crops across India.
- Create a mobile app.
- Can also add a speech recognition module to make the website more user friendly.

References

1. Beginning Django, Web Application Development and Deployment with Python @ Author (Rubio, Daniel) published 2001.
2. Learning PostgreSQL 10 @ Author (Salahaldin Juba and Andrey Volkov)
3. Django tutorials – <https://www.youtube.com/watch?v=OTmQOjsl0eg>