

21ECE371T – Database Design and Management
SEMESTER- V

Mini Project Report

ACADEMIC YEAR: 2024-25 ODD

Project Title: The Farm Management System

Name & Registration Number:

Aryan Kumar – RA2211053010019

Rachit Trivedi – RA2211053010046

Gaurav – RA2211053010047



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under SECTION 3 of the UGC Act, 1956)
S.R.M. NAGAR, KATTANKULATHUR – 603203.
CHENGALPATTU DISTRICT

NOVEMBER 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY College of Engineering and Technology School of Electrical and Electronics Department of Electronics and Communication Engineering
21ECC311L - Database Design and Management V Semester, 2024-2025 (ODD Semester)

Title of Mini Project : The Farm Management System

Date of Submission : 24/10/24

Particulars	Max. Marks	Marks Obtained		
		Name: Aryan Kumar	Name: Rachit Trivedi	Name: Gaurav
		Register No: RA2211053010019	Register No: RA2211053010046	Register No: RA2211053010047
Demonstration	05			
Viva	05			
Project Report	05			
Total	15			

REPORT VERIFICATION

Staff Name : Dr. V.Nithya

Signature :

The Farm Management System

1.Objective:

The Farm Management System is designed to be a centralized platform where farmers can efficiently manage farm activities and sell their products online, while buyers can browse and purchase agricultural goods directly from the system. By automating and centralizing tasks, the system reduces the manual processes farmers face and enhances communication between farmers and buyers, leading to a more streamlined marketplace for agricultural products.

2.Software Requirements:

Frontend- HTML, CSS, Java Script, Bootstrap

Backend-Python flask (Python 3.7) , SQLAlchemy,

- Operating System: Windows 10
- Google Chrome/Internet Explorer
- XAMPP (Version-3.7)
- Python main editor (user interface): PyCharm Community
- Workspace editor: Sublime text 3

3.Hardware Requirements:

- Computer with a 1.1 GHz or faster processor
- Minimum 2GB of RAM or more
- 2.5 GB of available hard-disk space
- 5400 RPM hard drive
- 1366 × 768 or higher-resolution display
- DVD-ROM drive

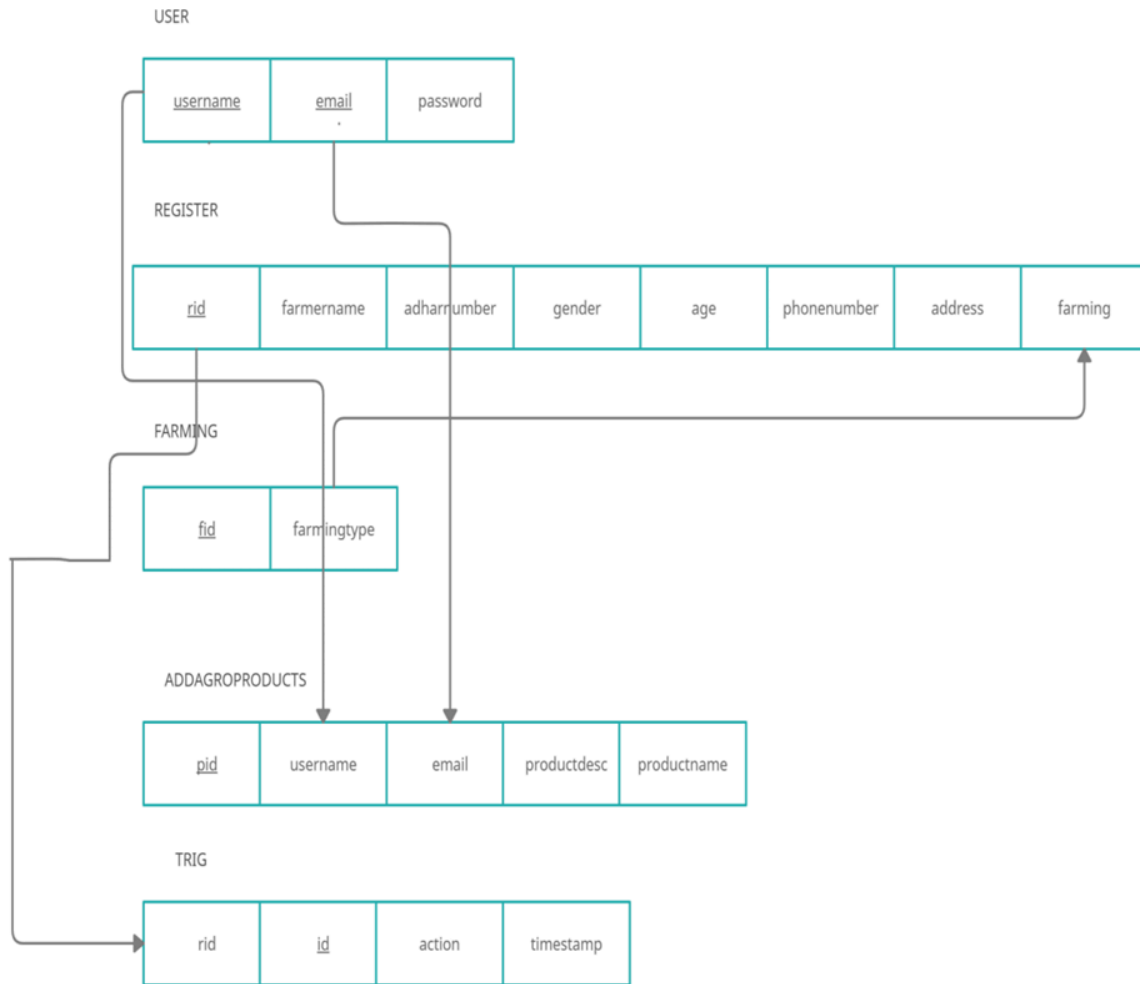
4.Introduction:

Our project, the Farm Management System, is a web-based application aimed at improving efficiency for both farmers and buyers in agricultural processes. It provides a platform for farmers to share best practices, sell agricultural products, and access farming tools and seeds. The goal is to modernize agriculture through digital means, making it easier for farmers to connect with buyers and manage their activities through an intuitive, user-friendly interface.

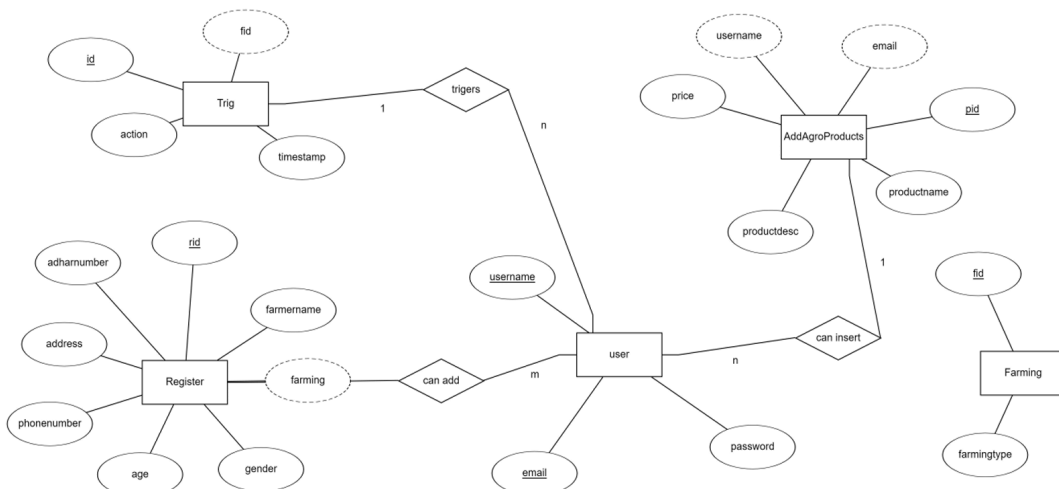
5.Limitations:

- Small size of farm business: Due to fragmentation and subdivision of holding the average size of operational holdings is very small
- Less labour per unit areas is required to farm large areas, especially since expensive alterations to land (like terracing) are completely absent.
- Mechanisation can be used more effectively over large, flat areas

6.Block Diagram:



7. ER Diagram:



8. Backend Python with MYSQL Code:

```
from flask import Flask,render_template,request,session,redirect,url_for,flash
from flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin
from werkzeug.security import generate_password_hash,check_password_hash
from flask_login import login_user,logout_user,login_manager,LoginManager
from flask_login import login_required,current_user
# MY db connection
local_server= True
app = Flask(__name__)
app.secret_key='harshithbhaskar'
# this is for getting unique user access
login_manager=LoginManager(app)
login_manager.login_view='login'
@login_manager.user_loader
def load_user(user_id):
return User.query.get(int(user_id))
app.config['SQLALCHEMY_DATABASE_URL']='mysql://username:password@localhost/data
bas_table_
name'
app.config['SQLALCHEMY_DATABASE_URI']='mysql://root:@localhost/farmers'
db=SQLAlchemy(app)
# here we will create db models that is tables
class Test(db.Model):
id=db.Column(db.Integer,primary_key=True)
name=db.Column(db.String(100))
class Farming(db.Model):
fid=db.Column(db.Integer,primary_key=True)
farmingtype=db.Column(db.String(100))

class Addagroproducts(db.Model):
username=db.Column(db.String(50))
email=db.Column(db.String(50))
pid=db.Column(db.Integer,primary_key=True)
productname=db.Column(db.String(100))
productdesc=db.Column(db.String(300))
price=db.Column(db.Integer)

class Trig(db.Model):
id=db.Column(db.Integer,primary_key=True)
fid=db.Column(db.String(100))
action=db.Column(db.String(100))
timestamp=db.Column(db.String(100))

class User(UserMixin,db.Model):
```

```
id=db.Column(db.Integer,primary_key=True)
username=db.Column(db.String(50))
email=db.Column(db.String(50),unique=True)
password=db.Column(db.String(1000))
```

```
class Register(db.Model):
    rid=db.Column(db.Integer,primary_key=True)
    farmername=db.Column(db.String(50))
    adharnumber=db.Column(db.String(50))
    age=db.Column(db.Integer)
    gender=db.Column(db.String(50))
    phonenumber=db.Column(db.String(50))
    address=db.Column(db.String(50))
    farming=db.Column(db.String(50))
```

```
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/farmerdetails')
```

```
@login_required
```

```
def farmerdetails():
    query=db.engine.execute(f"SELECT * FROM `register`")
    return render_template('farmerdetails.html',query=query)
```

```
@app.route('/agroproducts')
```

```
def agroproducts():
    query=db.engine.execute(f"SELECT * FROM `addagroproducts`")
    return render_template('agroproducts.html',query=query)
```

```
@app.route('/addagroproduct',methods=['POST','GET'])
```

```
@login_required
```

```
def addagroproduct():
    if request.method=="POST":
        username=request.form.get('username')
        email=request.form.get('email')
        productname=request.form.get('productname')
        productdesc=request.form.get('productdesc')
        price=request.form.get('price')
        products=Addagroproducts(username=username,email=email,productname=productname,productdesc=productdesc,price=price)
        db.session.add(products)
        db.session.commit()
        flash("Product Added","info")
        return redirect('/agroproducts')
    return render_template('addagroproducts.html')
@app.route('/triggers')
```

```

@login_required
def triggers():
    query=db.engine.execute(f'SELECT * FROM `trig`')
    return render_template('triggers.html',query=query)
@app.route('/addfarming',methods=['POST','GET'])
@login_required
def addfarming():
    if request.method=="POST":
        farmingtype=request.form.get('farming')
        query=Farming.query.filter_by(farmingtype=farmingtype).first()
        if query:
            flash("Farming Type Already Exist","warning")
            return redirect('/addfarming')
        dep=Farming(farmingtype=farmingtype)
        db.session.add(dep)
        db.session.commit()
        flash("Farming Addes","success")
    return render_template('farming.html')
@app.route("/delete/<string:rid>",methods=['POST','GET'])
@login_required
def delete(rid):
    db.engine.execute(f'DELETE FROM `register` WHERE `register`.`rid`={rid}')
    flash("Slot Deleted Successful","danger")
    return redirect('/farmerdetails')

@app.route("/edit/<string:rid>",methods=['POST','GET'])
@login_required
def edit(rid):
    farming=db.engine.execute("SELECT * FROM `farming`")
    posts=Register.query.filter_by(rid=rid).first()
    if request.method=="POST":
        farmername=request.form.get('farmername')
        adharnumber=request.form.get('adharnumber')
        age=request.form.get('age')
        gender=request.form.get('gender')
        phonenumber=request.form.get('phonenumber')

        address=request.form.get('address')
        farmingtype=request.form.get('farmingtype')
        query=db.engine.execute(f'UPDATE `register` SET
`farmername`='{farmername}',`adharnumber`='{adharnumber}',`age`='{age}',`gender`='{gender
}',`phonenu
mber`='{phonenumber}',`address`='{address}',`farming`='{farmingtype}'')
        flash("Slot is Updates","success")
        return redirect('/farmerdetails')
    return render_template('edit.html',posts=posts,farming=farming)

```

```

@app.route('/signup',methods=['POST','GET'])
def signup():
    if request.method == "POST":
        username=request.form.get('username')
        email=request.form.get('email')
        password=request.form.get('password')
        print(username,email,password)
        user=User.query.filter_by(email=email).first()
        if user:
            flash("Email Already Exist","warning")
            return render_template('/signup.html')
        encpassword=generate_password_hash(password)

        new_user=db.engine.execute(f"INSERT INTO `user` (`username`,`email`,`password`)
VALUES
('{username}','{email}','{encpassword}')")
        # this is method 2 to save data in db
        # newuser=User(username=username,email=email,password=encpassword)
        # db.session.add(newuser)
        # db.session.commit()
        flash("Signup Succes Please Login","success")
        return render_template('login.html')
    return render_template('signup.html')
@app.route('/login',methods=['POST','GET'])
def login():
    if request.method == "POST":
        email=request.form.get('email')
        password=request.form.get('password')
        user=User.query.filter_by(email=email).first()

        if user and check_password_hash(user.password,password):
            login_user(user)
            flash("Login Success","primary")
            return redirect(url_for('index'))
        else:
            flash("invalid credentials","danger")
            return render_template('login.html')

    return render_template('login.html')
app.route('/logout')
@login_required
def logout():
    logout_user()
    flash("Logout SuccessFul","warning")
    return redirect(url_for('login'))

```



```

@app.route('/register',methods=['POST','GET'])
@login_required
def register():
    farming=db.engine.execute("SELECT * FROM `farming`")
    if request.method=="POST":
        farmername=request.form.get('farmername')
        adharnumber=request.form.get('adharnumber')
        age=request.form.get('age')
        gender=request.form.get('gender')
        phonenumber=request.form.get('phonenumber')
        address=request.form.get('address')
        farmingtype=request.form.get('farmingtype')

        query=db.engine.execute(f"INSERT INTO `register`
('farmername`,`adharnumber`,`age`,`gender`,`phonenumber`,`address`,`farming`) VALUES
('{farmername}','{adharnumber}','{age}','{gender}','{phonenumber}','{address}','{farmingtype}'
)")
        flash("Your Record Has Been Saved","success")
        return redirect('/farmerdetails')
    return render_template('farmer.html',farming=farming)

@app.route('/test')
def test():
    try:
        Test.query.all()
        return 'My database is Connected'
    except:
        return 'My db is not Connected'

app.run(debug=True)

```

9. Frontend Code:

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <title>{% block title %}
    {% endblock title %}</title>
    <meta content="" name="description">
    <meta content="" name="keywords">
    {% block style %}
    {% endblock style %}

```

```

<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway
:300,400,50
0,700,800" rel="stylesheet">
<!-- Vendor CSS Files -->
<link href="static/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="static/assets/vendor/venobox/venobox.css" rel="stylesheet">
<link href="static/assets/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">
<link href="static/assets/vendor/owl.carousel/assets/owl.carousel.min.css" rel="stylesheet">
<link href="static/assets/vendor/aos/aos.css" rel="stylesheet">

<!-- Template Main CSS File -->
<link href="static/assets/css/style.css" rel="stylesheet">
</head>
<body>

<!-- ===== Header ===== -->
<header id="header">
  <div class="container">

    <div id="logo" class="pull-left">

      <a href="/" class="scrollto">F.M.S</a>
    </div>
    <nav id="nav-menu-container">
<ul class="nav-menu">
  <li class="{% block home %}
    {% endblock home %}"><a href="/">Home</a></li>
<li><a href="/register">Farmer Register</a></li>
<li><a href="/addfarming">Add Farming</a></li>
<li><a href="/farmerdetails">Farmer Details</a></li>
<li><a href="/agroproducts">Agro Products</a></li>
<li><a href="/triggers">Records</a></li>
  {% if current_user.is_authenticated %}
    <li class="buy-tickets"><a href="">Welcome {{current_user.username}}</a></li>
    <li class="buy-tickets"><a href="/logout">Logout</a></li>
  {% else %}
    <li class="buy-tickets"><a href="/signup">Signin</a></li>

  {% endif %}
</ul>
</nav><!-- #nav-menu-container -->
</div>
</header><!-- End Header -->

<!-- ===== Intro Section ===== -->

```

```

<section id="intro">
  <div class="intro-container" data-aos="zoom-in" data-aos-delay="100">
    <h1 class="mb-4 pb-0">SELL AGRO PRODUCTS AND BUY </span> </h1>
    <p class="mb-4 pb-0">DBMS Mini Project Using Flask & MYSQL</p>
    <a href="/agroproducts" class="about-btn scrollto">AGRO PRODUCTS</a>
  </div>
</section><!-- End Intro Section -->
<main id="main">
  {% block body %}
  {% with messages=get_flashed_messages(with_categories=true) %}
  {% if messages %}
  {% for category, message in messages %}

<div class="alert alert-{{category}} alert-dismissible fade show" role="alert">
  {{message}}
</div>
  {% endfor %}
  {% endif %}
  {% endwith %}
  {% endblock body %}

<a href="#" class="back-to-top"><i class="fa fa-angle-up"></i></a>

<!-- Vendor JS Files -->
<script src="static/assets/vendor/jquery/jquery.min.js"></script>
<script src="static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="static/assets/vendor/jquery.easing/jquery.easing.min.js"></script>
<script src="static/assets/vendor/php-email-form/validate.js"></script>
<script src="static/assets/vendor/venobox/venobox.min.js"></script>
<script src="static/assets/vendor/owl.carousel/owl.carousel.min.js"></script>
<script src="static/assets/vendor/superfish/superfish.min.js"></script>
<script src="static/assets/vendor/hoverIntent/hoverIntent.js"></script>
<script src="static/assets/vendor/aos/aos.js"></script>

<!-- Template Main JS File -->
<script src="static/assets/js/main.js"></script>
</body>
</html> <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>{% block title %}
  {% endblock title %}</title>
  <meta content="" name="description">

```

```

<meta content="" name="keywords">
{% block style %}
{% endblock style %}
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway
:300,400,50
0,700,800" rel="stylesheet">
<!-- Vendor CSS Files -->
<link href="static/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="static/assets/vendor/venobox/venobox.css" rel="stylesheet">
<link href="static/assets/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">
<link href="static/assets/vendor/owl.carousel/assets/owl.carousel.min.css" rel="stylesheet">
<link href="static/assets/vendor/aos/aos.css" rel="stylesheet">

<!-- Template Main CSS File -->
<link href="static/assets/css/style.css" rel="stylesheet">
</head>
<body>

<!-- ===== Header ===== -->
<header id="header">
  <div class="container">
<div id="logo" class="pull-left">
  <a href="/" class="scrollto">F.M.S</a>
</div>
  <nav id="nav-menu-container">
    <ul class="nav-menu">
      <li class="{% block home %}">
        {% endblock home %} "><a href="/">Home</a></li>
<li><a href="/register">Farmer Register</a></li>
<li><a href="/addfarming">Add Farming</a></li>
<li><a href="/farmerdetails">Farmer Details</a></li>
<li><a href="/agroproducts">Agro Products</a></li>
<li><a href="/triggers">Records</a></li>
      {% if current_user.is_authenticated %}
        <li class="buy-tickets"><a href="">Welcome {{current_user.username}}</a></li>
        <li class="buy-tickets"><a href="/logout">Logout</a></li>
      {% else %}
        <li class="buy-tickets"><a href="/signup">Signin</a></li>

      {% endif %}
    </ul>
  </nav><!-- #nav-menu-container -->
</div>
</header><!-- End Header -->

```

```

<!-- ===== Intro Section ===== -->
<section id="intro">
    <div class="intro-container" data-aos="zoom-in" data-aos-delay="100">
        <h1 class="mb-4 pb-0">SELL AGRO PRODUCTS AND BUY </span> </h1>
        <p class="mb-4 pb-0">DBMS Mini Project Using Flask & MYSQL</p>
        <a href="/agroproducts" class="about-btn scrollTo">AGRO PRODUCTS</a>
    </div>
</section><!-- End Intro Section -->
<main id="main">

```

```

{% block body %}

```

```

{% with messages=get_flashed_messages(with_categories=true) %}
{% if messages %}
{% for category, message in messages %}
<div class="alert alert-{{category}} alert-dismissible fade show" role="alert">
    {{message}}
</div>
{% endfor %}
{% endif %}
{% endwith %}
{% endblock body %}

```

```

<a href="#" class="back-to-top"><i class="fa fa-angle-up"></i></a>
<!-- Vendor JS Files -->
<script src="static/assets/vendor/jquery/jquery.min.js"></script>
<script src="static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="static/assets/vendor/jquery.easing/jquery.easing.min.js"></script>
<script src="static/assets/vendor/php-email-form/validate.js"></script>
<script src="static/assets/vendor/venobox/venobox.min.js"></script>
<script src="static/assets/vendor/owl.carousel/owl.carousel.min.js"></script>
<script src="static/assets/vendor/superfish/superfish.min.js"></script>
<script src="static/assets/vendor/hoverIntent/hoverIntent.js"></script>
<script src="static/assets/vendor/aos/aos.js"></script>
<!-- Template Main JS File -->
<script src="static/assets/js/main.js"></script>
</body>
</html>

```

Farmer.html

```

{% extends 'base.html' %}
{% block title %}
Add Farming
{% endblock title %}

```

```

{% block body %}
<h3 class="text-center bg-success text-white"><span>Add Farming</span> </h3>

{% with messages=get_flashed_messages(with_categories=true) %}
{% if messages %}
{% for category, message in messages %}

<div class="alert alert-{{category}} alert-dismissible fade show" role="alert">
    {{message}}
</div>
{% endfor %}
{% endif %}
{% endwith %}
<br>
<div class="container">
<div class="row">
<div class="col-md-4"></div>
<div class="col-md-4">
<form action="/addfarming" method="post">

<div class="form-group">
<label for="dept">Enter Farming Type</label>
<input type="text" class="form-control" name="farming" id="farming">
</div>
<br>
    <button type="submit" class="btn btn-success btn-sm btn-block">Add Farming</button>
</form>
<br>
<br>
</div>
<div class="col-md-4"></div>
</div></div>
{% endblock body %}

```

10. ScreenShots:

Farm Management

SIGN UP

UserName

Email address

We'll never share your email with anyone else.

Password

Sign In

Already User?

Login

F.M.S

Home Farmer Register Add Farming Farmer Details Agro Products Records Welcome test Logout

Agro Products

GIRIJA CAULIFLOWER
Price : 520

Tips for Growing Cauliflower.
Well drained medium loam and
or sandy loam soils are suitable.

Owner : test

Email : test@gmail.com

Purchase

COTTON
Price : 563

Cotton is a soft, fluffy staple fiber
that grows in a boll, around the
seeds of the cotton

Owner : test

Email : test@gmail.com

Purchase

silk
Price : 582

silk is best business developed
from cocoon for saries
preparation and so on

Owner : arkpro

Email : arkpro@gmail.com

Purchase

Agriculture Management System | localhost / 127.0.0.1 / farmers | Add AgroProducts

127.0.0.1:5000/addagroproduct

Apps | Configuring G Suite... | intoDNS: arkprocod... | How To Serve Djan... | localhost / 127.0.0... | MVPS | godaddy email | DOWNLOAD CV | Django Final Year P...

F.M.S | Home | Farmer Register | Add Farming | Farmer Details | Agro Products | Records | Welcome test | Logout

Farmer Email
test@gmail.com

Product Name
cotton

Product Description
make silk sarees

Price
500

Add Product

localhost / 127.0.0.1 / farmers | phpMyAdmin

localhost/phpmyadmin/sql.php?server=1&db=farmers&table=register&pos=0

Server: 127.0.0.1 | Database: farmers | Table: register

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | More

Showing rows 0 - 0 (1 total, Query took 0.0019 seconds)

SELECT * FROM 'register'

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	rid	farmername	adharnumber	age	gender	phonenumber	address	farming
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	mohit	8574857485748574	22	male	9986786453	banaglore	Seed Farming

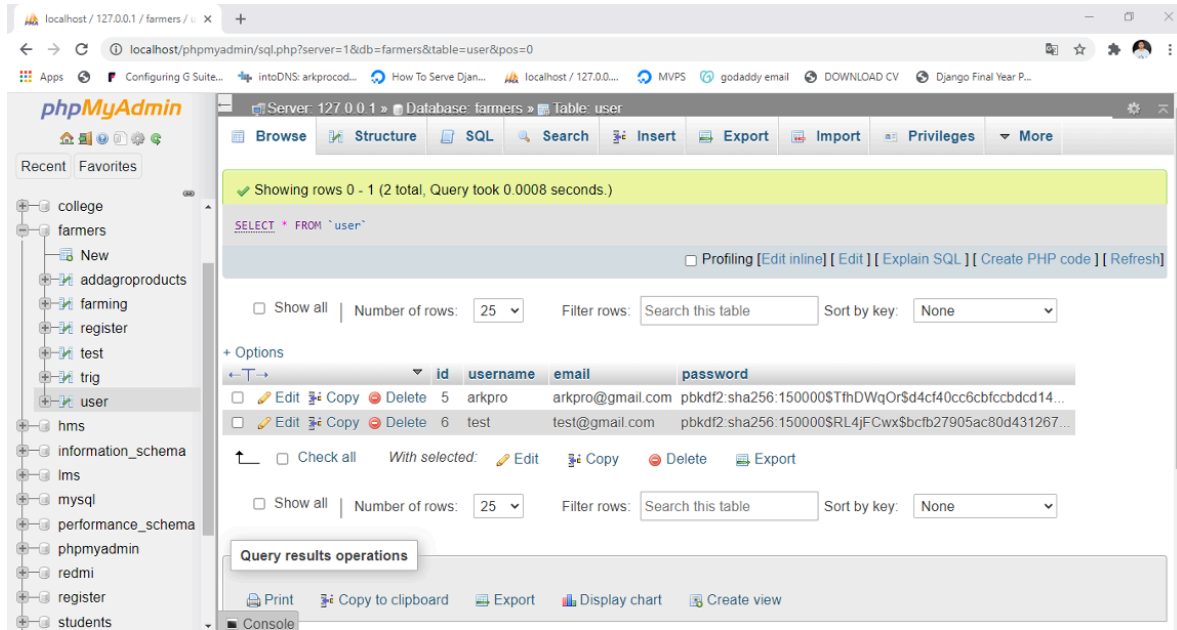
Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console



11. Future Enhancement:

- Enhanced database storage facility
- Enhanced user friendly GUI
- more advanced results systems
- online payments

12. Conclusion:

The Farm Management System project was a valuable learning experience that successfully integrated theory and practice. The system has streamlined the online selling of agro-products, providing farmers with a simple, efficient way to manage and sell their products. Using tools like Python Flask, XAMPP, and MySQL, we created a fully functional platform for managing farm-related data and facilitating transactions. MySQL, being free and highly customizable, allowed us to efficiently store, retrieve, and manipulate data according to the system's requirements.

This project has reinforced key concepts such as the importance of careful planning, the value of structured methodologies, and the power of teamwork in bringing a project from concept to completion. It provided practical insights into database management, web development, and the coordination required to create a fully functioning application, making us more competent as software engineers.