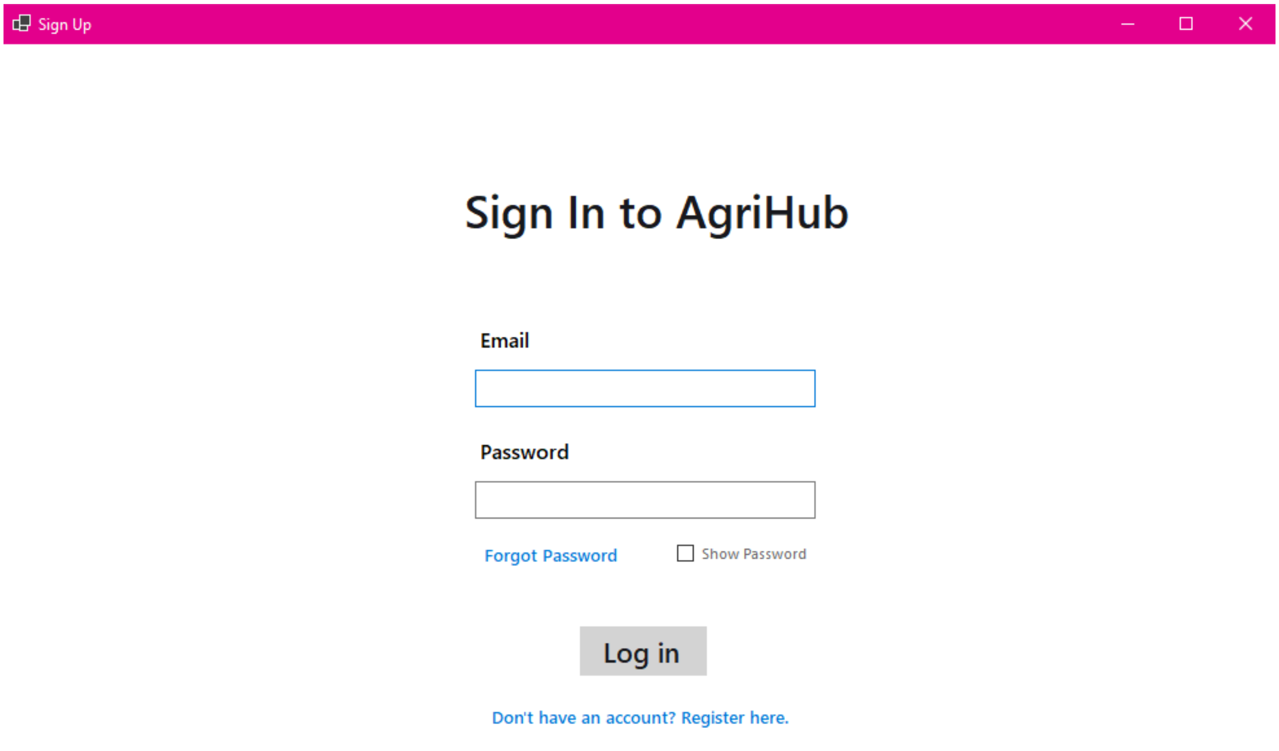
CSC104-Database Systems
Final Project Report(Spring 2025)

Student Details

Repository Name	DBS25F020
Registration Number	2024-CS-06, 2024-CS-16, 2024-CS-59

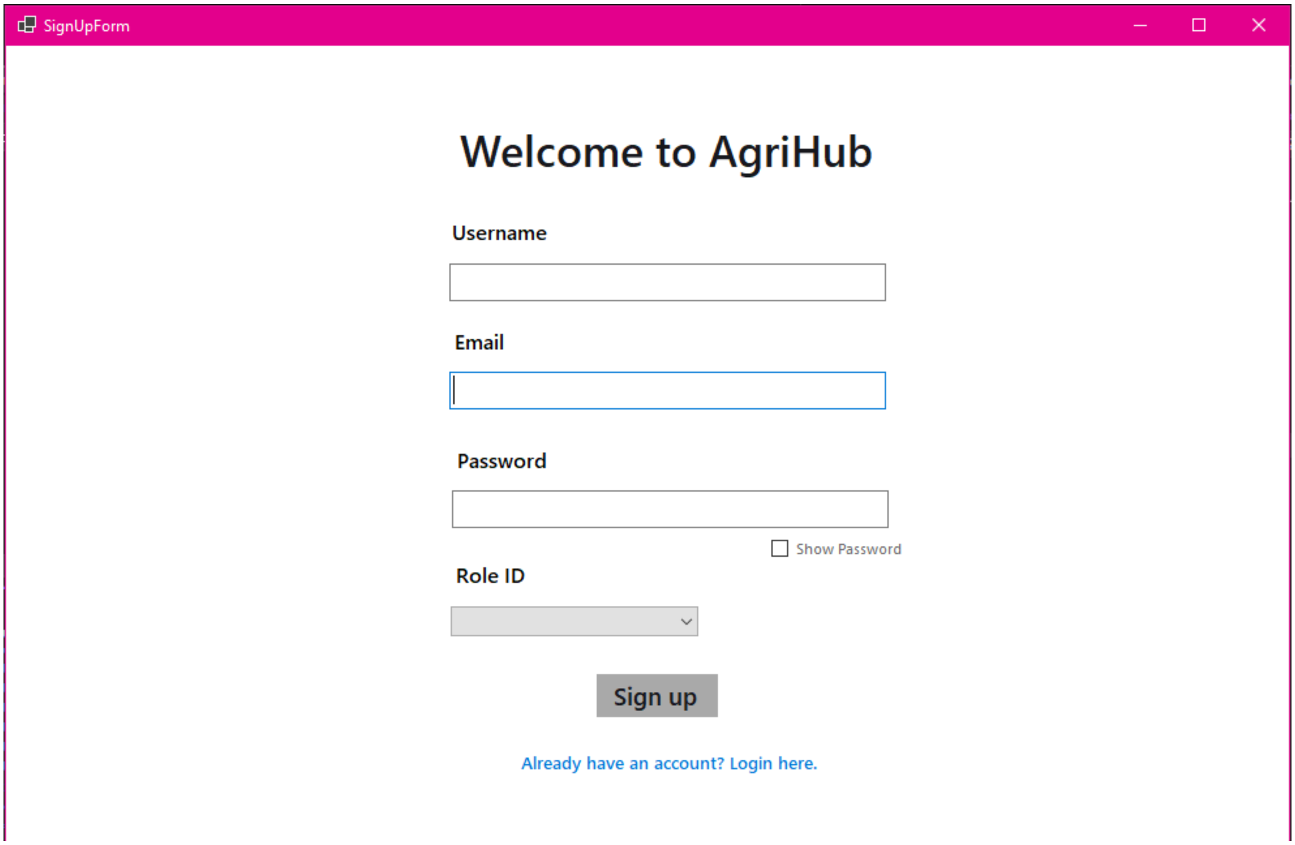
Project	
Project Title	**Agriculture Management System**
Executive Summary	This project is about to create an easy to use system for farmers which helps in farming and managing the lands. It is made as a Window Forms application using C# and to store all the information it is connected with the MySQL Database. This system is designed in 3 tier model i.e User Interface, Business Layer, Database Layer. The purpose of this project is to help the farmers to manage their records, farms, crops, fertilizers and stock. Farmer can login and use it to see his farms information, he can add the crops he planted, make a record of his equipment and look after the products in stock. He can consult Experts in any regards. This system can also be used by Experts to give response to farmer questions. Application has a simple design. Grids are used to show data about farms, crops, stock etc. It also helps farmer to perform CRUD operations on many products. Farmers can also manage the water usage, resources like tube wells or canals, and keep track of how much usage hour of the land. It also helps farmer to manage fertilizers and other agricultural products like pesticides, seeds. Farmer enters the quantity he used, price of product and for which crop. It has many features but one of them is its ability to generate reports in PDF. These reports contain the data entered by farmer and saved for generating reports in future. The system also supports searching and filtering of files and records specially in grids farmer can search all he want in this way farmer can easily find what they need. Everything is connected to eachother and forms are shown in a way which is very easy to understand. The aim is to support small and big farmers by giving them a proper digital record-keeping managemnt system.

Login Form	



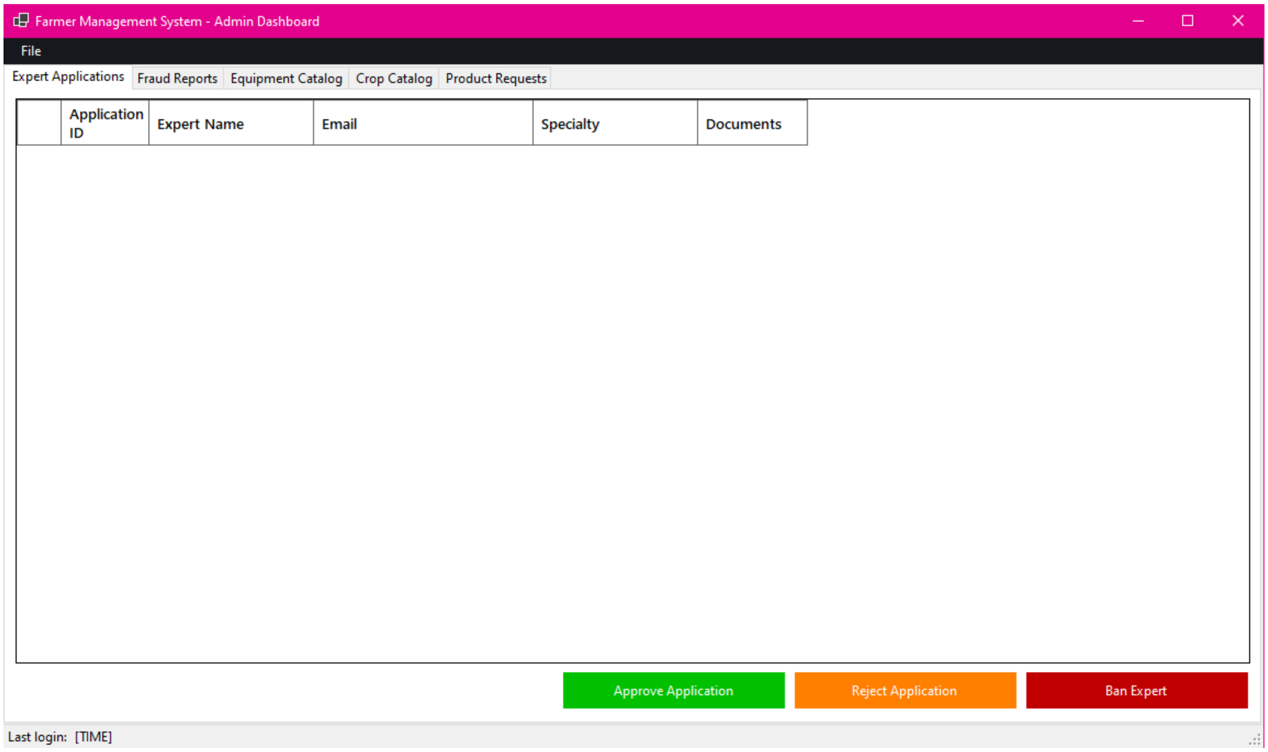
UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
txtUsername	TextBox	LoginUsername	users (username)
txtPassword	TextBox	LoginPassword	users (password)
btnLogin	Button	ValidateLogin()	SELECT * FROM users WHERE username=@Username AND password=@Password
lblForgotPassword	Label	OpenPasswordRecovery	N/A
lblTitle	Label	LoginFormTitle	N/A

SignUp Form	



UI Component Name	**Type of UI Component**	**Class & Attribute Name**	**Linked Database Table/Column**
txtEmail	TextBox	User.Email	users.email
btnVerifyEmail	Button	VerifyEmail()	users (SELECT * FROM users WHERE email = @Email)
txtNewPassword	TextBox	User.Password	users.password
btnResetPassword	Button	ResetPassword()	users (UPDATE users SET password = @NewPassword WHERE email = @Email)
label4	Label	FormTitle	N/A

Admin Form



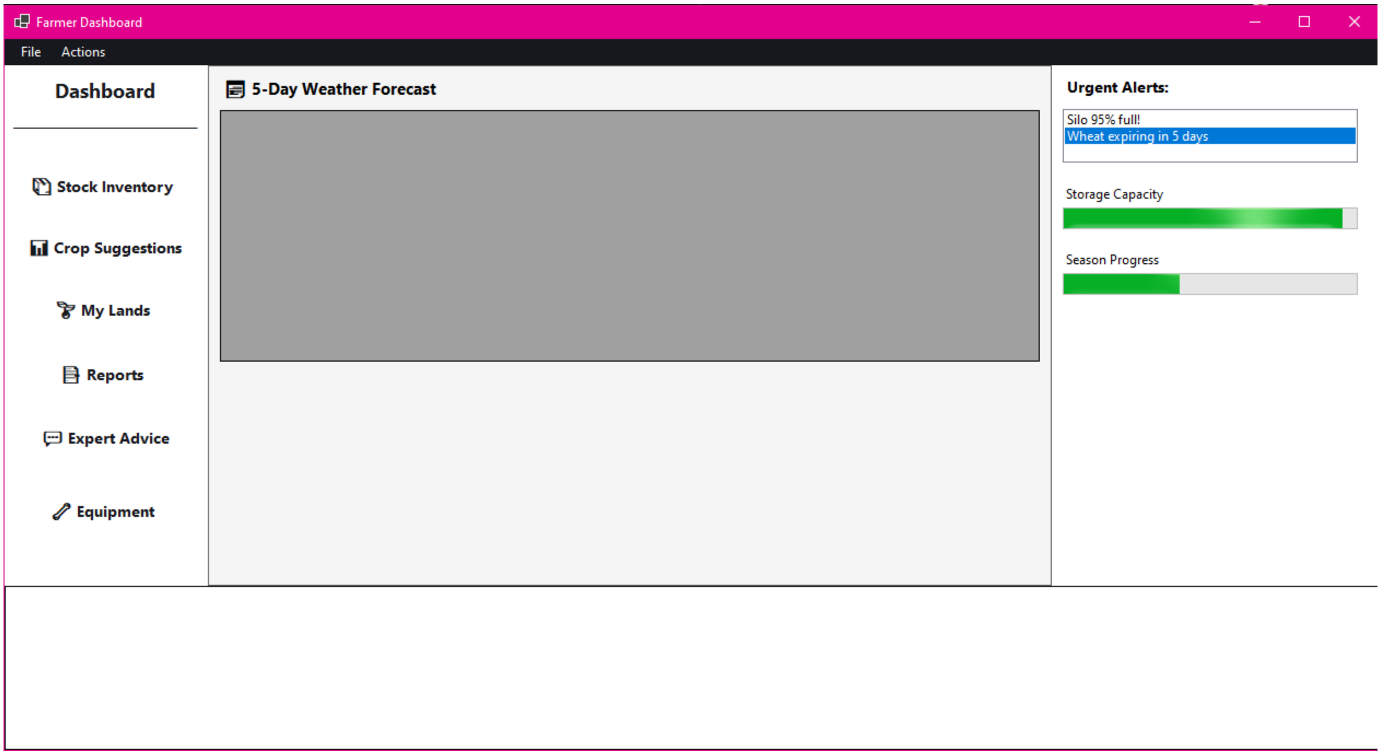
	Application ID	Expert Name	Email	Specialty	Documents

Last login: [TIME]

UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
dgvExpertApps	DataGridView	dgvExpertApplications	expert_applications (SELECT id, name, email, specialty FROM expert_applications WHERE status='Pending')
dgvFraudReports	DataGridView	dgvFraudReports	fraud_reports (SELECT id, reporter_id, expert_id, report_type FROM fraud_reports)
dgvEquipment	DataGridView	dgvEquipment	equipment (SELECT id, name FROM equipment)
dgvCrops	DataGridView	dgvCrops	crops (SELECT id, name FROM crops)
dgvProductRequests	DataGridView	dgvProductRequests	product_requests (SELECT id, farmer_id, product_type FROM product_requests)

In case of column, write column name and in case of multiple columns, write query in last column

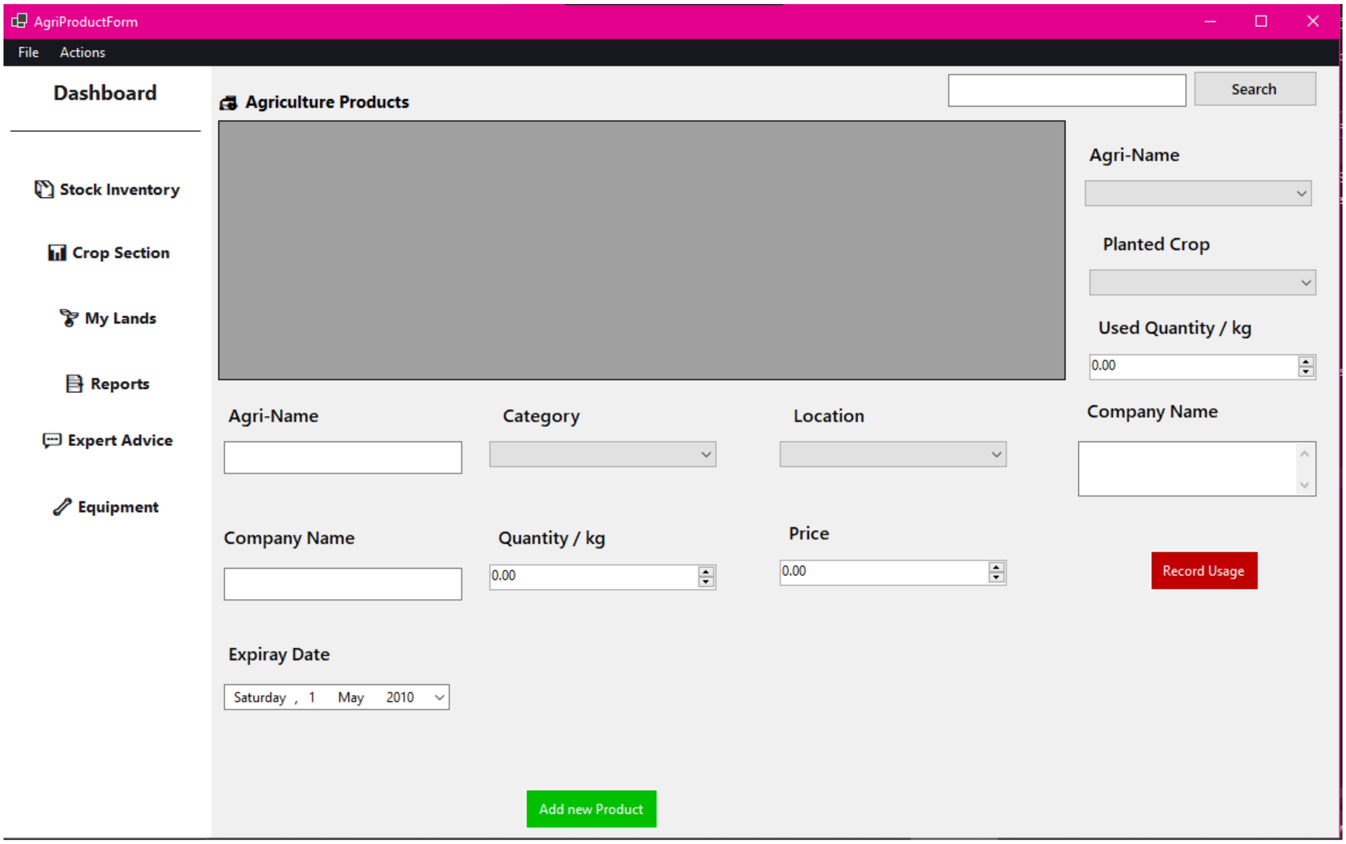
FarmerDashboard Form



UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
;dgvWeatherForecast	DataGridView	WeatherForecast	weather_data
lstAlerts	ListBox	AlertList	
progressStorage	ProgressBar	StorageCapacity	inventory (SELECT SUM(quantity)/SUM(capacity) FROM storage)
progressSeason	ProgressBar	SeasonProgress	N/A
dataGridViewRecent	DataGridView	RecentActivities	activities (SELECT * FROM activities ORDER BY date DESC LIMIT 10)

In case of column, write column name and in case of multiple columns, write query in last column

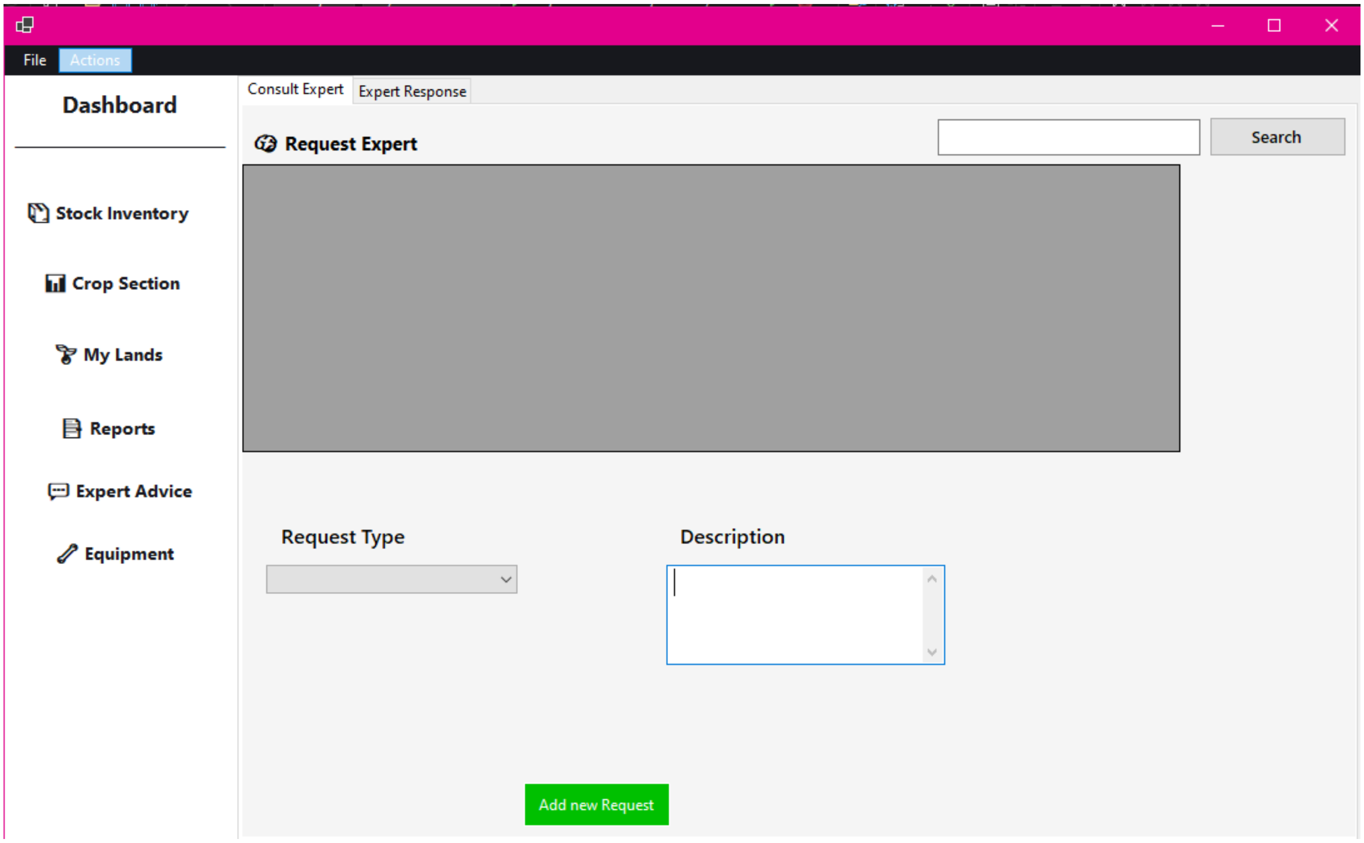
Agri_Products Form



UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
dgvAgriProducts	DataGridView	grdstock	agri_products (SELECT id, name, quantity FROM agri_products WHERE farmer_id=@id)
cmbCategory	ComboBox	comboBox1	product_categories (SELECT DISTINCT category FROM

			agri_products)
txtProductName	TextBox	txtagriname	agri_products.name
txtManufacturer	TextBox	txtcompany	agri_products.manufacturer
numQuantity	NumericUpDown	numericUpDown3	agri_products.quantity
numPrice	NumericUpDown	numericUpDown1	agri_products.price
dtpExpiry	DateTimePicker	Expiry	agri_products.expiry_date
btnAddProduct	Button	btnaddagri	agri_products (INSERT)

Expert Advice Form

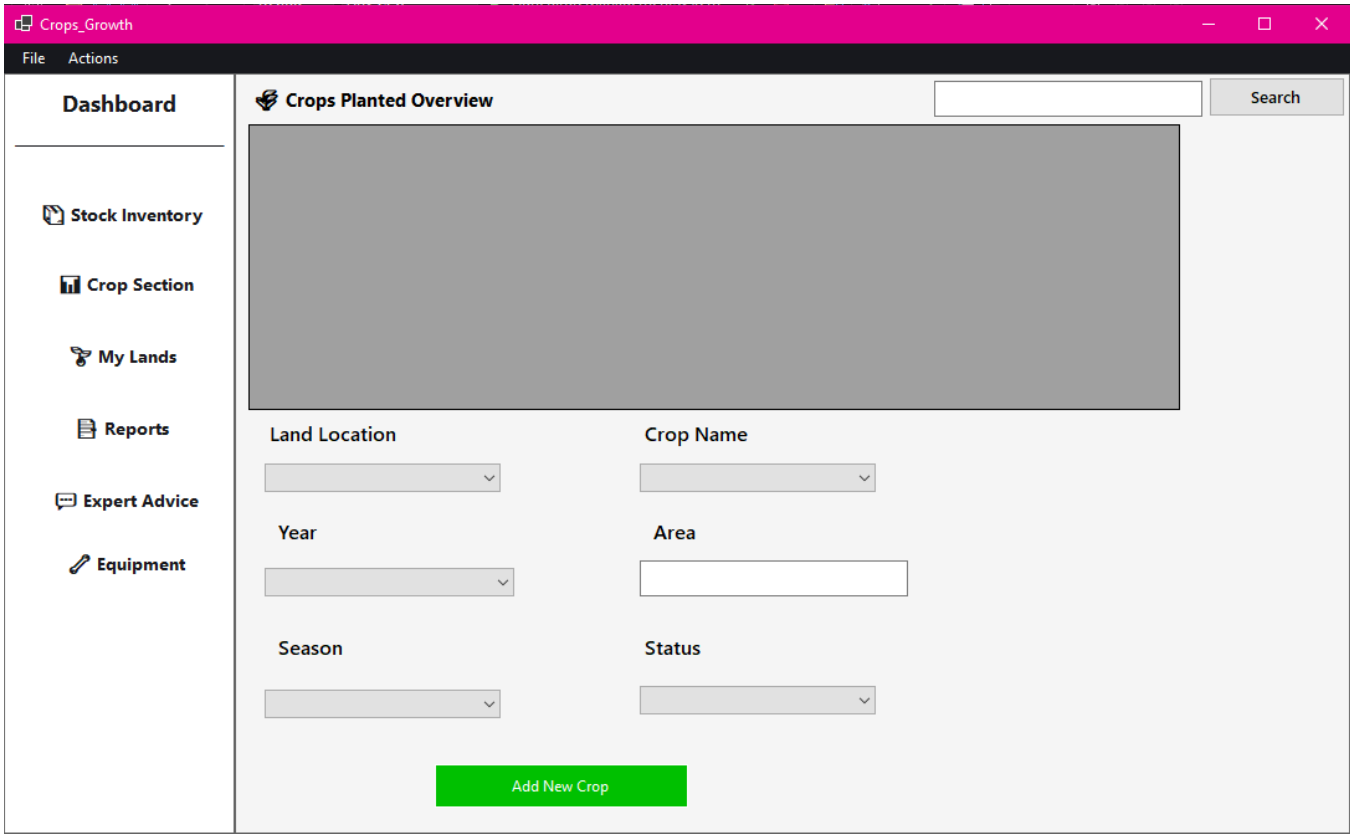


UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
tabControlExpert	TabControl	ConsultationTabs	N/A
tabRequest	TabPage	RequestTab	N/A
txtSearch	TextBox	SearchRequests	requests (SELECT * FROM requests WHERE title LIKE '%@Search%')
btnSearch	Button	SearchRequests()	SELECT * FROM requests
grdrequest	DataGridView	RequestList	SELECT * FROM requests
cmbpurpose	ComboBox	RequestPurpose	N/A

cmbrequestype	ComboBox	RequestType	N/A
btnaddrequest	Button	AddRequest()	INSERT INTO requests
tabexpertresponse	TabPage	ResponseTab	N/A
txtResponse	TextBox	ExpertResponse	N/A
btnSubmitResponse	Button	SubmitResponse()	UPDATE requests SET response=@Response

In case of column, write column name and in case of multiple columns, write query in last column

Crop Form

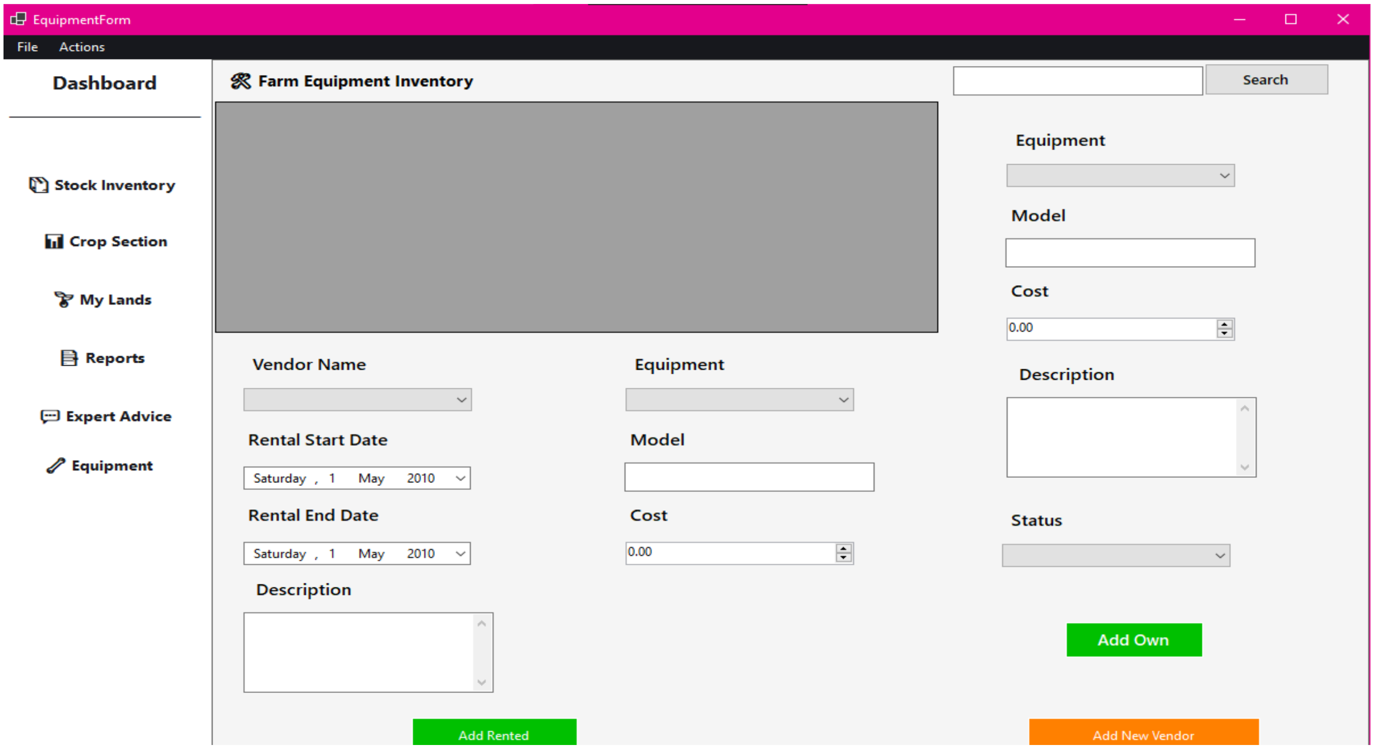


UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
txtSearch	TextBox	SearchCrops	crops (SELECT * FROM crops WHERE name LIKE '%@Search%')
btnSearch	Button	SearchCrops()	SELECT * FROM crops
grdlands	DataGridView	CropsList	SELECT * FROM crops
comboBox1	ComboBox	CropName	crops (name)
comboBox2	ComboBox	CropStatus	N/A

comboBox3	ComboBox	Season	N/A
comboBox4	ComboBox	Year	N/A
txtarea	TextBox	CropArea	N/A
btnAddCrop	Button	AddCrop()	INSERT INTO crops

In case of column, write column name and in case of multiple columns, write query in last column

Equipment Form

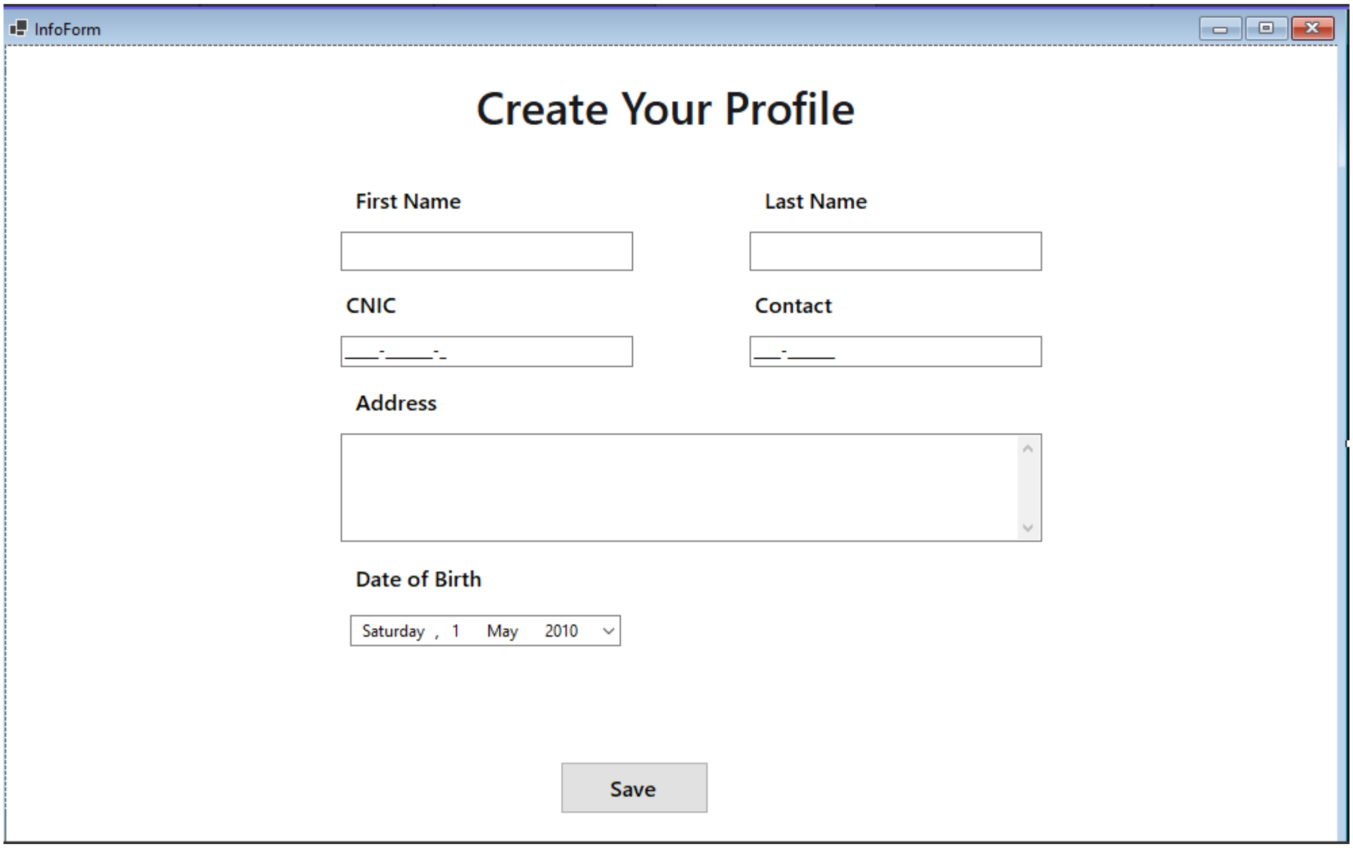


UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
txtSearch	TextBox	SearchEquipment	equipment (SELECT * FROM equipment WHERE name LIKE '%@Search%')
btnSearch	Button	SearchEquipment()	SELECT * FROM equipment
grdEquipments	DataGridView	EquipmentList	SELECT * FROM equipment
cmbequipment	ComboBox	EquipmentType	equipment_types
cmbStatus	ComboBox	EquipmentStatus	N/A
cmbvendorname	ComboBox	VendorName	vendors
cmbrentequipment	ComboBox	RentalEquipment	equipment
btnnewvendor	Button	AddVendor()	INSERT INTO vendors

btnowned	Button	AddOwnedEquipment()	INSERT INTO equipment

In case of column, write column name and in case of multiple columns, write query in last column

Info Form



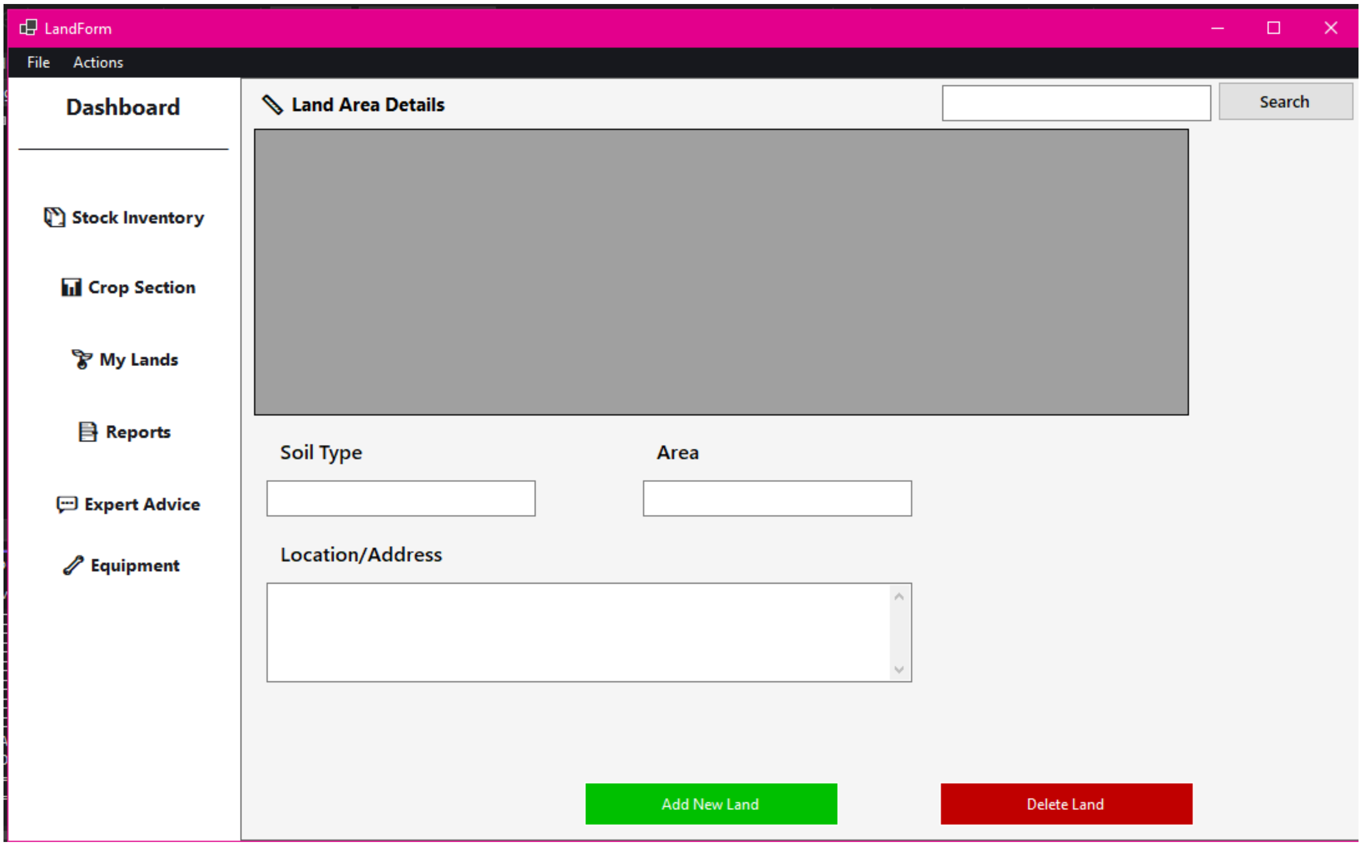
Date of Birth

Saturday , 1 May 2010

UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
lblTitle	Label	label4	Experts.full_name
txtFirstName	TextBox	txtFname	Experts.first_name
txtLastName	TextBox	txtLname	Experts.last_name
mtxtCNIC	MaskedTextBox	CNICmask	Experts.cnic
mtxtContact	MaskedTextBox	Contactmask	Experts.contact_number
dtpDOB	DateTimePicker	Dob	Experts.date_of_birth
txtAddress	TextBox	txtAdress	Experts.address
btnSave	Button	btnSave	Experts (INSERT/UPDATE)

In case of column, write column name and in case of multiple columns, write query in last column

Land Form

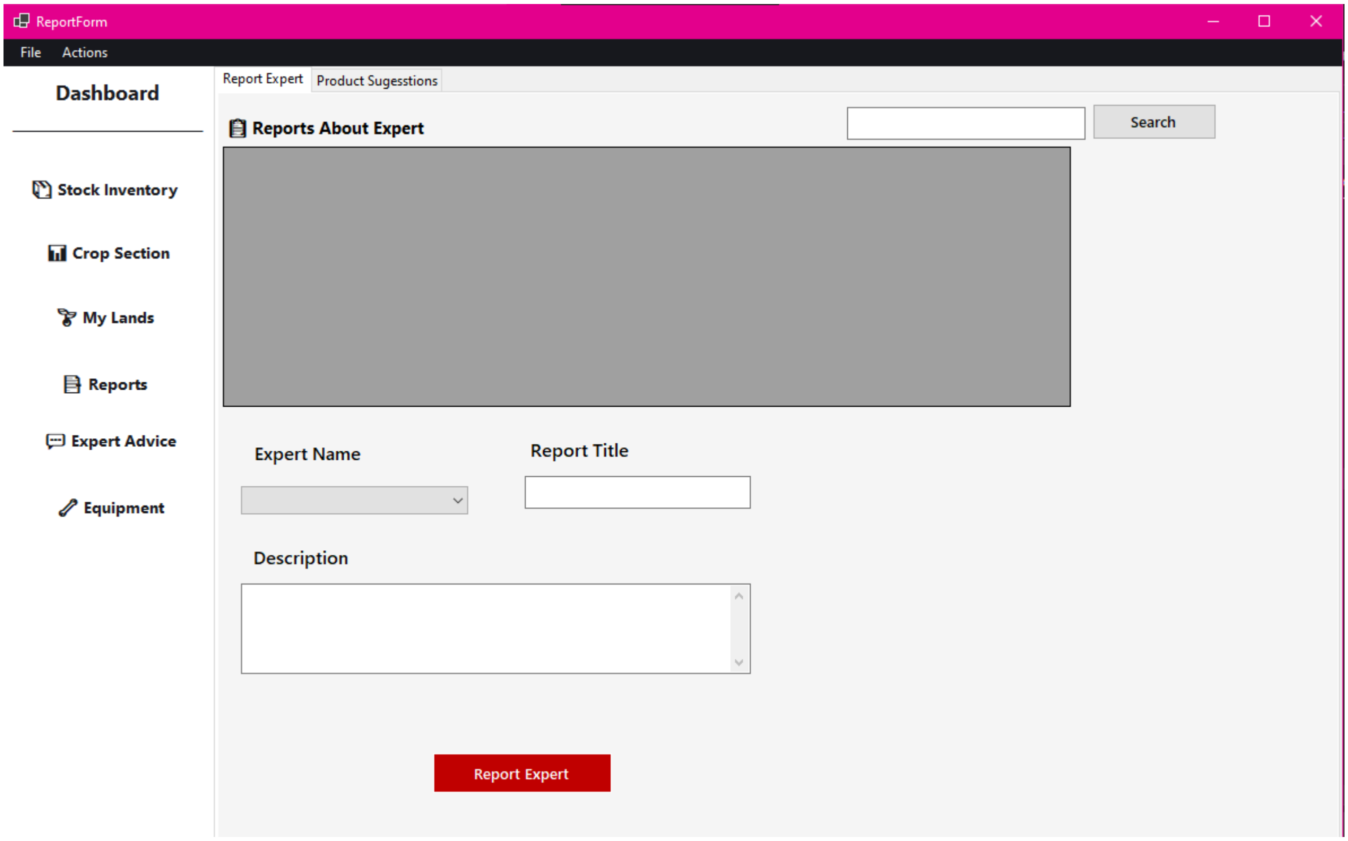


UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
txtSearch	TextBox	SearchLands	lands (SELECT * FROM lands WHERE location LIKE '%@Search%')
btnSearch	Button	SearchLands()	SELECT * FROM lands
grdlands	DataGridView	LandList	SELECT * FROM lands
txtsoil	TextBox	SoilType	N/A
txtarea	TextBox	LandArea	N/A
txtAddress	TextBox	LandAddress	N/A
btnAddLand	Button	AddLand()	INSERT INTO lands

btnDeleteLand	Button	DeleteLand()	DELETE FROM lands WHERE id=@ID

In case of column, write column name and in case of multiple columns, write query in last column

Request Resources Form

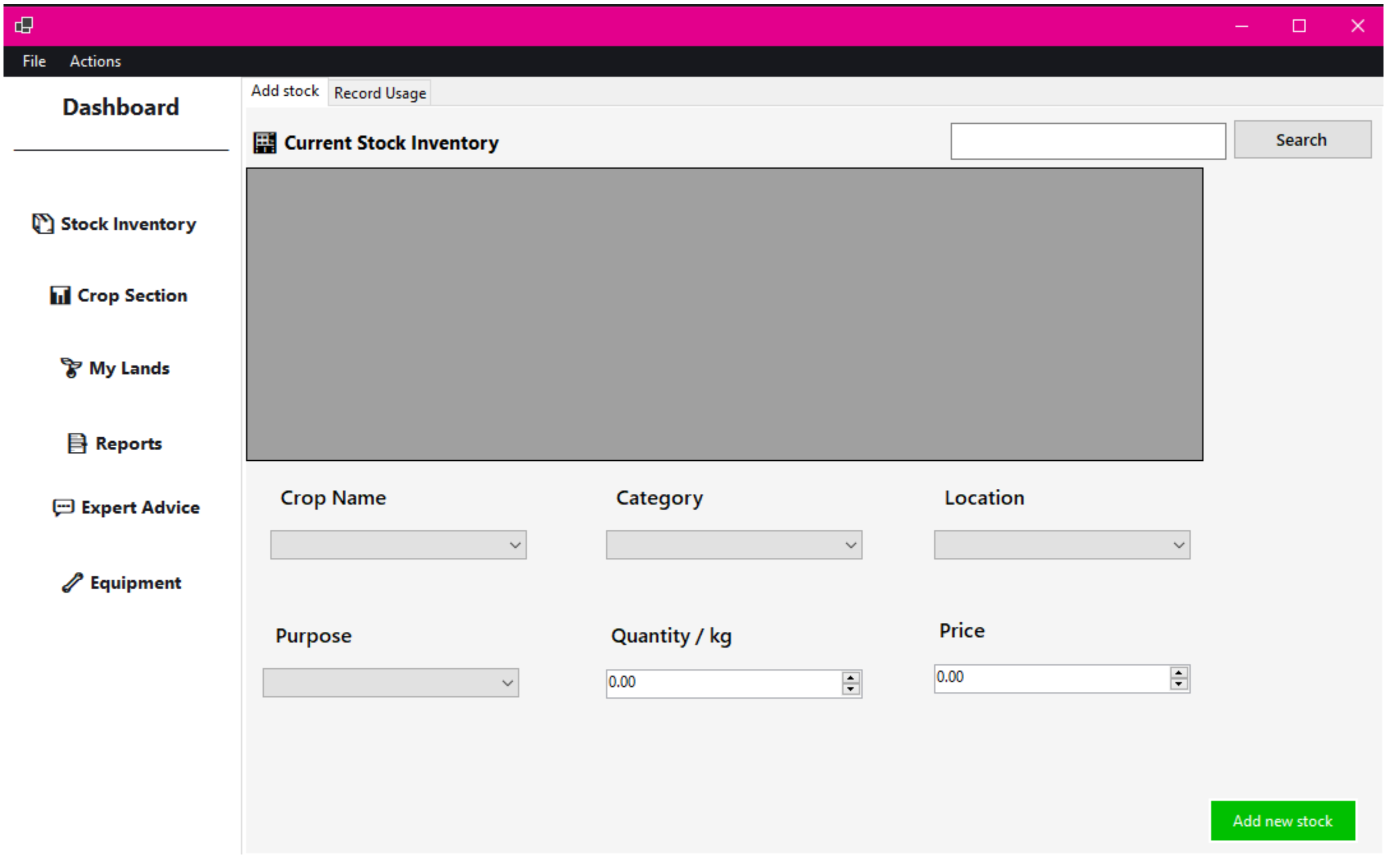


UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
tabControlExpert	TabControl	ReportTabs	N/A
tabReport	TabPage	ReportTab	N/A
txtSearch	TextBox	SearchReports	reports (SELECT * FROM reports WHERE title LIKE '%@Search%')
btnSearch	Button	SearchReports()	SELECT * FROM reports
grdstock	DataGridView	ReportList	SELECT * FROM reports

cmbcrop	ComboBox	ExpertName	experts
txttitle	TextBox	ReportTitle	N/A
txtdes	TextBox	ReportDescription	N/A
btnReport	Button	SubmitReport()	INSERT INTO reports
tabsuggestion	TabPage	SuggestionTab	N/A
textBox3	TextBox	SuggestionTitle	N/A
textBox2	TextBox	SuggestionDescription	N/A
btnsuggest	Button	SubmitSuggestion()	INSERT INTO suggestions

In case of column, write column name and in case of multiple columns, write query in last column

Stock Inventory Form

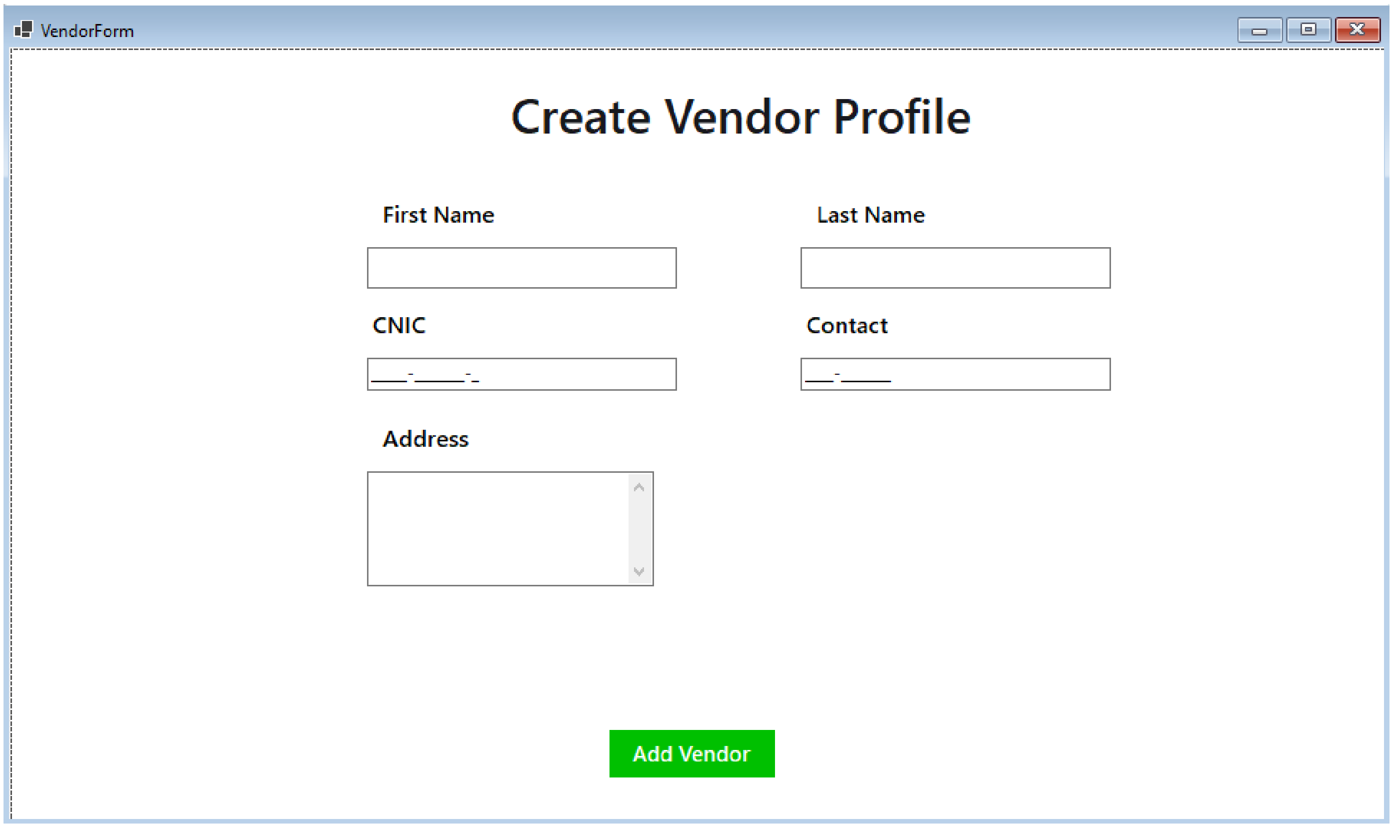


UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
dgvInventory	DataGridView	grdstock	inventory (SELECT id, product_name, quantity FROM inventory WHERE farmer_id=@id)

cmbCrop	ComboBox	cmbcrop	crops (SELECT id, name FROM crops)
numStockQuantity	NumericUpDown	numericUpDown1	inventory.quantity
btnAddStock	Button	btnaddstock	inventory (INSERT)

In case of column, write column name and in case of multiple columns, write query in last column

Vendor Profile Form

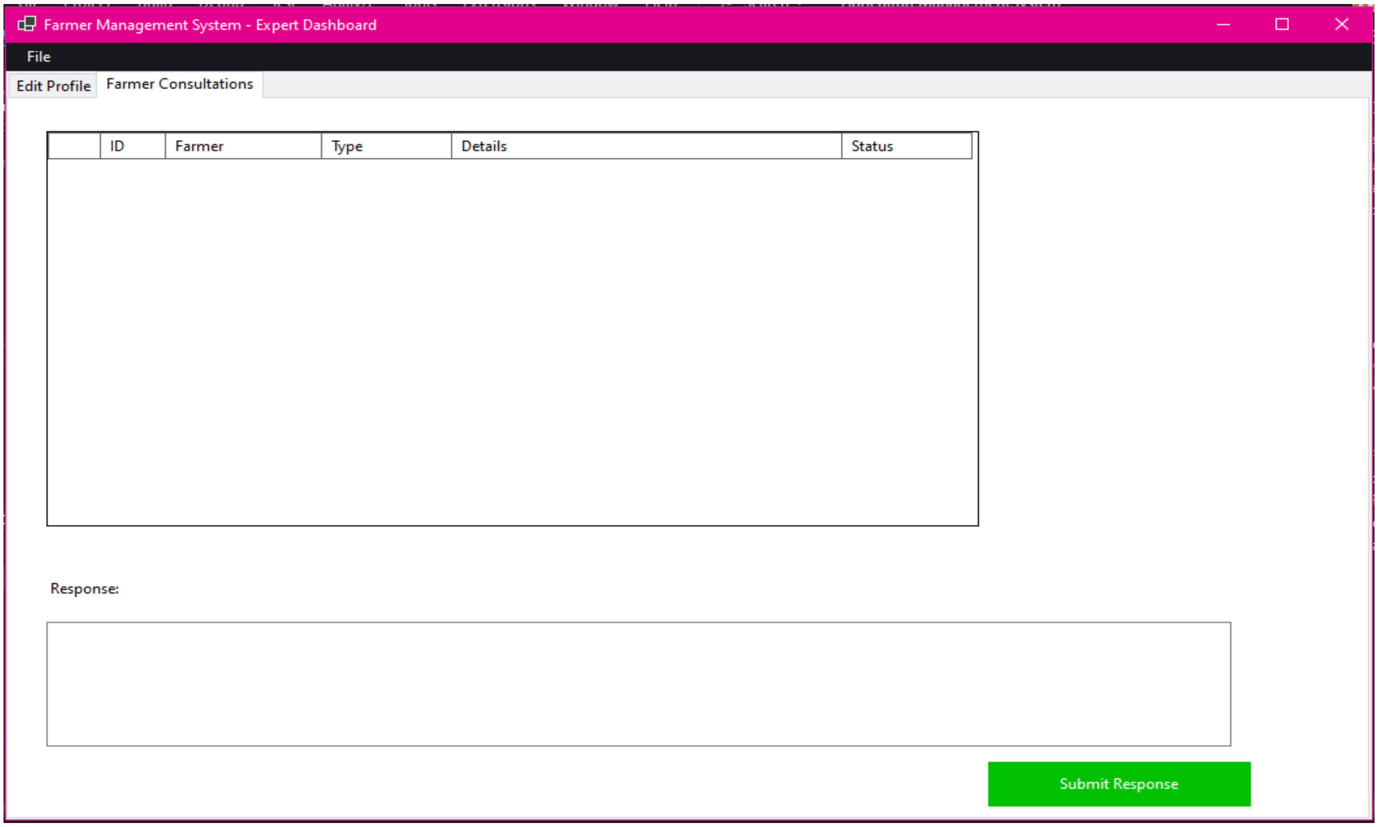


Create Vendor Profile

UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
txtVendorFirstName	TextBox	txtFname	vendors.first_name
txtVendorLastName	TextBox	txtLname	vendors.last_name
mtxtVendorCNIC	MaskedTextBox	CNICmask	vendors.cnic
mtxtVendorPhone	MaskedTextBox	Contactmask	vendors.phone
txtVendorAddress	TextBox	txtAddress	vendors.address
btnAddVendor	Button	btnaddvendor	vendors (INSERT)

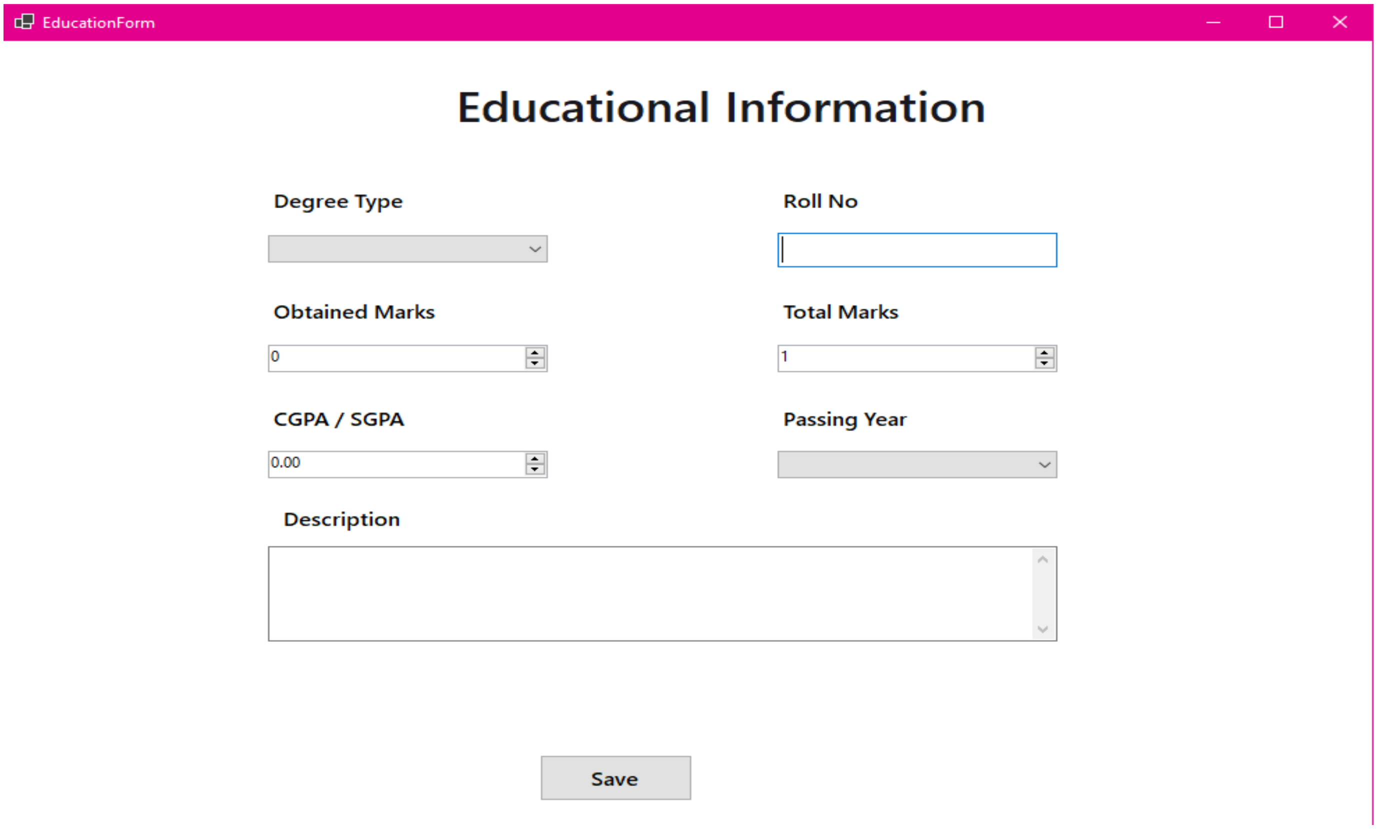
In case of column, write column name and in case of multiple columns, write query in last column

Expert Dashboard Form



UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
dgvConsultations	DataGridView	gridFramerRequest	consultations (SELECT id, farmer_name, request_type, details FROM consultations WHERE expert_id=@id)
txtExpertResponse	TextBox	txtResponse	consultations.response
btnSubmitResponse	Button	btnSubmitResponse	consultations (UPDATE response WHERE id=@selected_id)

Educational Info Form



EducationForm

Educational Information

Degree Type:

Roll No:

Obtained Marks: 0

Total Marks: 1

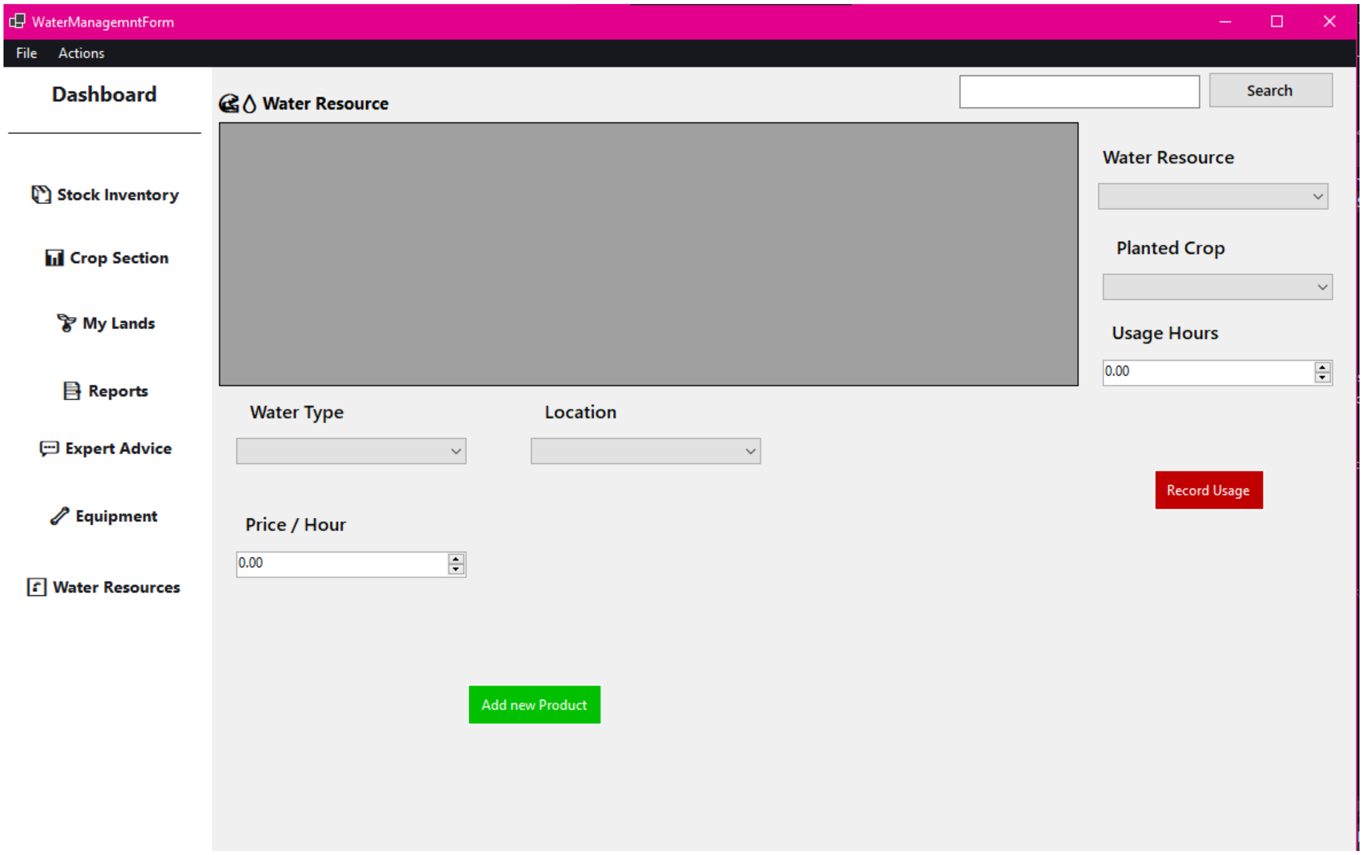
CGPA / SGPA: 0.00

Passing Year:

Description:

UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
cmbDegreeType	ComboBox	comboBox1	Education.degree_type
txtRollNo	TextBox	txtEmail	Education.roll_number
numObtainedMarks	NumericUpDown	numericUpDown1	Education.obtained_marks
numTotalMarks	NumericUpDown	numericUpDown2	Education.total_marks
numCGPA	NumericUpDown	numericUpDown3	Education.gpa
cmbPassingYear	ComboBox	comboBox2	Education.passing_year
txtDescription	TextBox	textBox1	Education.description
btnSaveEducation	Button	button3	Education (INSERT/UPDATE)

Water Resources Form



UI Component Name	Type of UI Component	Class & Attribute Name	Linked Database Table/Column
txtSearch	TextBox	SearchWaterResources	water_resources (SELECT * FROM water_resources WHERE type LIKE '%@Search%')
btnSearch	Button	SearchWaterResources( )	SELECT * FROM water_resources
grdstock	DataGridView	WaterResourceList	SELECT * FROM water_resources
cmbwater	ComboBox	WaterType	N/A
cmbplanted	ComboBox	PlantedCrop	crops
numericUpDown5	NumericUpDown	UsageHours	N/A
btnusage	Button	RecordUsage()	INSERT INTO water_usage
btnaddagri	Button	AddWaterResource()	INSERT INTO water_resources

Business Report-1	
Business Report Name:	*Water Consumption Report*
Sample of report:	

Query to build Report:

```
SELECT  p.planted_id,  p.crop_type,   p.field_location,
SUM(wu.usage_hours) AS total_water_hours,  SUM(wu.total_cost) AS total_water_cost,
AVG(wu.usage_hours) AS avg_daily_hours
FROM
    water_usage wu
JOIN
    planted_areas p ON wu.planted_id = p.planted_id
WHERE
    wu.usage_date BETWEEN @start_date AND @end_date
GROUP BY
    p.planted_id, p.crop_type, p.field_location
ORDER BY
    total_water_hours DESC;
```

Business Report-2	
Business Report Name:	*Crop Yield Report*
Sample of report:	

Query to build Report

```
SELECT
    p.planted_id,
    p.crop_type,
    p.field_location,
    p.planting_date,
    p.harvest_date,
    p.expected_yield,
    p.actual_yield,
    (p.actual_yield / NULLIF(p.expected_yield, 0)) * 100 AS yield_percentage,
    SUM(wu.usage_hours) AS total_water_hours,
    SUM(wu.total_cost) AS total_water_cost
FROM
    planted_areas p
LEFT JOIN
    water_usage wu ON p.planted_id = wu.planted_id
GROUP BY
    p.planted_id, p.crop_type, p.field_location, p.planting_date,
    p.harvest_date, p.expected_yield, p.actual_yield
ORDER BY
    yield_percentage DESC;
```

Business Report-3	
Business Report Name:	*Crop Production Report*
Sample of report:	

Query to build Report

```
SELECT fp.farmer_id, CONCAT(fp.first_name, ' ', fp.last_name) AS
farmer_name, pc.p_id AS planting_id, c.crop_name, pc.year, pc.season, pc.area AS planted_area,
SUM(wu.usage_hours) AS total_water_hours, SUM(wu.total_cost) AS water_cost, cb.quantity AS
harvested_quantity, cb.price AS market_price, (cb.quantity * cb.price) AS estimated_revenue
FROM
    farmer_profile fp
JOIN
    planted_crops pc ON fp.farmer_id = pc.farmer_id
JOIN
    crop c ON pc.crop_id = c.crop_id
LEFT JOIN
    water_usage wu ON pc.p_id = wu.planted_id
LEFT JOIN
    crop_batch cb ON cb.purpose = 'Harvested'
    AND cb.batch_id IN (SELECT crop_batch_id FROM crop_stock WHERE crop_name = c.crop_name)
WHERE
```

```
    pc.year = YEAR(CURDATE())
GROUP BY
    fp.farmer_id, pc.p_id, c.crop_id, cb.batch_id
ORDER BY
    estimated_revenue DESC;                    WHEN 2 THEN 'Fulfilled' END AS Status
                    FROM faculty_requests
                    JOIN faculty ON faculty_requests.faculty_id = faculty.faculty_id
                    JOIN consumables ON faculty_requests.item_id = consumables.consumable_id;
```

Business Report-4	
Business Report Name:	Rental Equipments Report
Sample of report:	

Query to build Report

```
SELECT
    v.vendor_id,
    CONCAT(v.first_name, ' ', v.last_name) AS vendor_name,
    e.eq_name AS equipment_type,
    COUNT(re.rental_id) AS rental_count,
    SUM(re.rental_cost) AS total_rental_income,
    AVG(DATEDIFF(re.rental_end_date, re.rental_start_date)) AS avg_rental_duration,
    MIN(re.rental_cost / DATEDIFF(re.rental_end_date, re.rental_start_date)) AS min_daily_rate,
    MAX(re.rental_cost / DATEDIFF(re.rental_end_date, re.rental_start_date)) AS max_daily_rate,
    AVG(re.rental_cost / DATEDIFF(re.rental_end_date, re.rental_start_date)) AS avg_daily_rate
FROM
    vendor_profile v
JOIN
    rented_equipment re ON v.vendor_id = re.vendor_id
JOIN
    equipments e ON re.eq_id = e.eq_id
WHERE
    re.rental_start_date BETWEEN DATE_SUB(CURDATE(), INTERVAL 1 YEAR) AND CURDATE()
GROUP BY
    v.vendor_id, e.eq_id
ORDER BY
    total_rental_income DESC;
```

Business Report-5	

Business Report Name: *profit Report*

Sample of report:

Query to build Report:

```
SELECT
    fp.farmer_id,
    CONCAT(fp.first_name, ' ', fp.last_name) AS farmer_name,

    SUM(cb.quantity * cb.price) AS total_income,

    SUM(wu.total_cost) AS total_water_cost,

    SUM(re.rental_cost) AS total_equipment_rental,

    SUM(ab.price_per_unit * au.used_quantity) AS total_agri_cost,

    SUM(cb.quantity * cb.price)
    - SUM(wu.total_cost)
    - SUM(re.rental_cost)
    - SUM(ab.price_per_unit * au.used_quantity) AS net_profit

FROM farmer_profile fp

LEFT JOIN planted_crops pc ON fp.farmer_id = pc.farmer_id
LEFT JOIN water_usage wu ON wu.planted_id = pc.p_id

LEFT JOIN rented_equipment re ON re.farmer_id = fp.farmer_id

LEFT JOIN agri_usage au ON au.planting_id = pc.p_id
LEFT JOIN agri_products ap ON ap.agr_id = au.agr_id
LEFT JOIN agri_batch ab ON ab.batch_id = ap.batch_id
LEFT JOIN crop_batch cb ON cb.purpose = 'Sold'

LEFT JOIN crop_stock cs ON cs.crop_batch_id = cb.batch_id


GROUP BY

    fp.farmer_id, fp.first_name, fp.last_name;
```