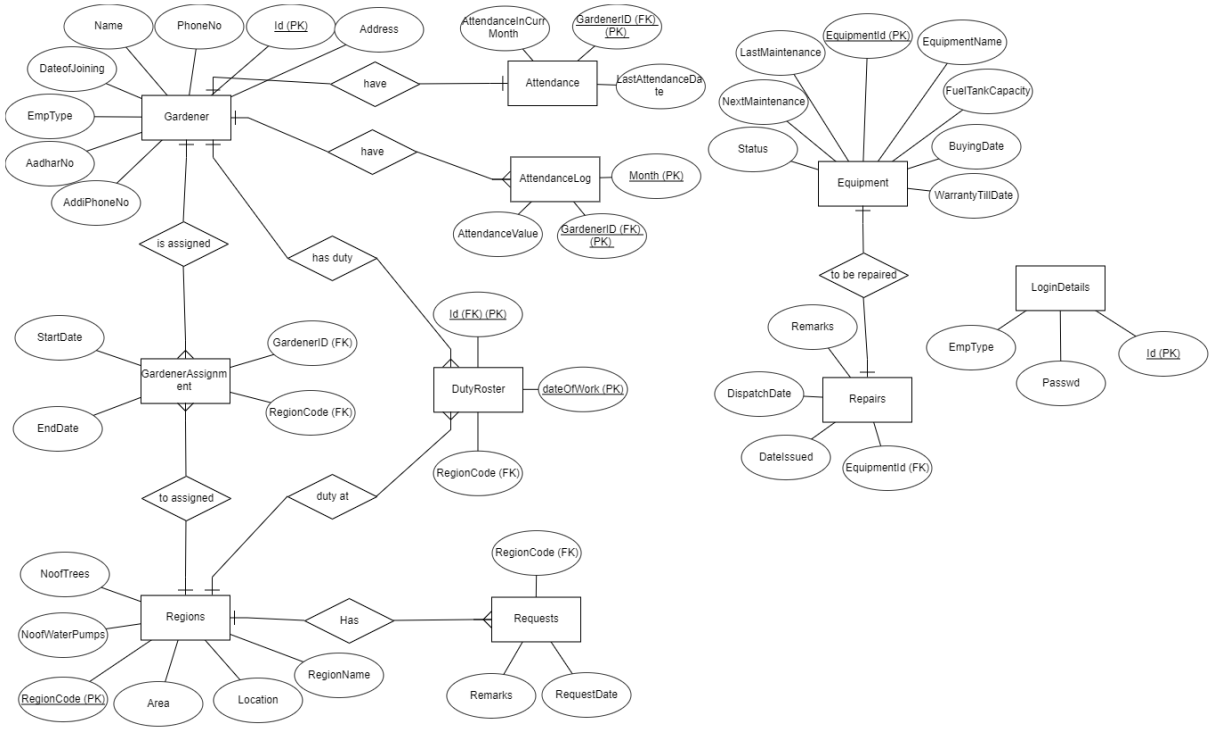
Landscaping Related Service Report

Name: Shubhajeet Dey (Team 6)
RollNo: 1901CS58
Files Link: Here. You can also load the sql txt file to form database.
ER DIAGRAM:



*** FK = Foreign Key, PK = Primary Key**

Database Tables Created:

LoginDetails:

This Table contains Login ID and Password details, Password is kept hashed here with SHA256 Algorithm.

```
CREATE TABLE LoginDetails(
Id varchar(8) PRIMARY KEY,
Passwd varchar(255) NOT NULL,
EmpType varchar(10) NOT NULL,
CONSTRAINT chk_EmpType_login CHECK (EmpType IN ('Supervisor', 'Gardener'))
);
```

Gardener:

This Table contains all details of the respective Gardener, Id is the primary key here. It also contains checks for Phone Number and Aadhar Number and Employee Type(Permanent, Temporary).

```
CREATE TABLE Gardener(
Id varchar(8) PRIMARY KEY,
Name varchar(30) NOT NULL,
PhoneNo char(10) NOT NULL,
Address varchar(50) NOT NULL,
AddiPhoneNo char(10),
AadharNo char(12) NOT NULL,
EmpType char(1) NOT NULL,
DateofJoining date NOT NULL,
CONSTRAINT chk_PhoneNo CHECK (PhoneNo NOT LIKE '%[^0-9]%'),
CONSTRAINT chk_AddiPhoneNo CHECK ((AddiPhoneNo IS NULL) OR (AddiPhoneNo NOT LIKE '%[^0-9]%')),
CONSTRAINT chk_AadharNo CHECK (AadharNo NOT LIKE '%[^0-9]%'),
CONSTRAINT chk_EmpType CHECK (EmpType IN ('P','T'))
);
```

Regions:

This Table contains all the details of respective regions in a large area, RegionCode is the primary key here. It also contains checks for Area, Number of Trees and Number of Water Pumps.

```
CREATE TABLE Regions(
RegionCode varchar(5) PRIMARY KEY,
RegionName varchar(30) NOT NULL,
Location varchar(50) NOT NULL,
Area double NOT NULL,
NoofTrees int,
NoofWaterPumps int,
CONSTRAINT chk_Area CHECK (Area > 0.0),
CONSTRAINT chk_NoofTrees CHECK ((NoofTrees IS NULL) OR (NoofTrees >= 0)),
CONSTRAINT chk_NoofWaterPumps CHECK ((NoofWaterPumps IS NULL) OR (NoofWaterPumps >= 0))
);
```

GardenerAssignment:

This Table contains details about assignments of Gardener in a particular area. It contains all the previous records, that is why there is no primary key as foreign key values can get repeated. To find current assignments, CURDATE() < EndDate.

```
CREATE TABLE GardenerAssignment(
GardenerID varchar(8) NOT NULL,
RegionCode varchar(5) NOT NULL,
StartDate  date NOT NULL,
EndDate    date NOT NULL,
CONSTRAINT Id_fk FOREIGN KEY (GardenerID) REFERENCES Gardener(Id),
CONSTRAINT RegionCode_fk FOREIGN KEY (RegionCode) REFERENCES Regions(RegionCode)
);
```

Attendance:

This Table contains Attendance details for current month, GardernerID is primary key as well as foreign key from Gardener Table. It also contains Last Attendance Date(last date attendance was marked).

```
CREATE TABLE Attendance(
GardenerID varchar(8) NOT NULL PRIMARY KEY,
LastAttendanceDate date NOT NULL,
AttendanceInCurrMonth int NOT NULL DEFAULT 0,
CONSTRAINT GardenerID_fk_attend FOREIGN KEY (GardenerID) REFERENCES Gardener(Id)
);
```

AttendanceLog:

This Table contains Log details of Attendance for each and every month with Month column in format (YYYY-MM). Here the combination of GardenerID and Month are taken as the primary key.

```
CREATE TABLE AttendanceLog(
GardenerID varchar(8) NOT NULL,
Month varchar(8) NOT NULL,
AttendanceValue int NOT NULL,
CONSTRAINT GardenerID_fk_attend_log FOREIGN KEY (GardenerID) REFERENCES Gardener(Id),
CONSTRAINT attendlog_pk PRIMARY KEY (GardenerID, Month)
);
```

DutyRoster:

This table contains duty roster details in the form of combination of Id, RegionCode, dateOfWork. Here combination of Id, dateOfWork is taken as primary key which means a Gardener can be assigned only one region at a date.

```
CREATE TABLE DutyRoster(
Id varchar(8) NOT NULL,
RegionCode varchar(5) NOT NULL,
dateOfWork date NOT NULL,
CONSTRAINT Id_fk_roster FOREIGN KEY (Id) REFERENCES Gardener(Id),
CONSTRAINT RegionCode_fk_roster FOREIGN KEY (RegionCode) REFERENCES Regions(RegionCode),
CONSTRAINT roster_pk PRIMARY KEY (Id, dateOfWork)
);
```

Requests:

This table contains all the grass cutting requests made with individual remarks and request date. A RegionCode can have several grass cutting requests.

```
CREATE TABLE Requests(
RegionCode varchar(5) NOT NULL,
Remarks varchar(100),
RequestDate date NOT NULL,
CONSTRAINT RegionCode_request_fk FOREIGN KEY (RegionCode) REFERENCES Regions(RegionCode)
);
```

Equipment:

This table contains equipment details. It also contains their maintenance details(upcoming maintenance date, previous maintenance date and warranty details). Checks are to ensure correct working of the database.

```
CREATE TABLE Equipment(
EquipmentId int AUTO_INCREMENT PRIMARY KEY,
EquipmentName varchar(40) NOT NULL,
FuelTankCapacity double,
BuyingDate date NOT NULL,
WarrantyTillDate date,
LastMaintenance date,
NextMaintenance date,
Status varchar(16) NOT NULL,
CONSTRAINT chk_FuelTankCapacity CHECK ((FuelTankCapacity IS NULL) OR (FuelTankCapacity > 0.0)),
CONSTRAINT chk_Status CHECK (Status IN ('INUSE','INREPAIR'))
);
```

Repairs:

This table contains details about repair of equipment(DispatchDate: Approx. Date till repaired, Remarks: Notes about Problems in the machinery, DateIssued: Date of issuance of repair).

```
CREATE TABLE Repairs(
EquipmentId int NOT NULL,
DateIssued date NOT NULL,
DispatchDate date,
Remarks varchar(150),
CONSTRAINT EquipmentId_fk FOREIGN KEY (EquipmentId) REFERENCES Equipment(EquipmentId),
);
```

Procedures used:

mark_attendance:

This procedure marks attendance by taking ID and a value as inputs(present=0: Absent, present=1: Present). The ID was taken in, check if it exists in attendance table or not, if not, then a new row is inserted in attendance table, otherwise the matching row gets updated with present value and new LastAttendanceDate.

```
-- Procedure to mark attendance (absent/present)

DELIMITER $$

CREATE PROCEDURE mark_attendance(in ID varchar(8), in present int)
BEGIN
DECLARE cnt int;
SELECT COUNT(*) INTO cnt FROM Attendance WHERE Attendance.GardenerID = ID;
IF (cnt > 0) THEN
    IF (present > 0) THEN
    UPDATE Attendance SET LastAttendanceDate = CURDATE(), AttendanceInCurrMonth = AttendanceInCurrMonth + 1 WHERE Attendance.GardenerID = ID;
    ELSE
    UPDATE Attendance SET LastAttendanceDate = CURDATE() WHERE Attendance.GardenerID = ID;
    END IF;
ELSE
    IF (present > 0) THEN
    INSERT INTO Attendance VALUES (ID, CURDATE(), 1);
    ELSE
    INSERT INTO Attendance VALUES (ID, CURDATE(), 0);
    END IF;
END IF;
END $$

Query OK, 0 rows affected (0.03 sec)

DELIMITER ;
```

submit_attendance:

This procedure submits the current month's attendance list to the attendance log table and resets the attendance count to 0 for the next month's attendance counting.

```
-- Procedure to submit/store attendances of current month in attendance log table and reset attendance count.

DELIMITER $$

CREATE PROCEDURE submit_attendance()
BEGIN
INSERT INTO AttendanceLog SELECT GardenerID, LEFT(LastAttendanceDate,7), AttendanceInCurrMonth FROM Attendance;
UPDATE Attendance SET AttendanceInCurrMonth = 0;
END $$

--Query OK, 0 rows affected (0.01 sec)

DELIMITER ;
```

thirty_days_from_now:

This procedure creates a DATE_TEMP table which is then loaded with next 30 day's dates (including today's date). This DATE_TEMP Table is then used in Duty Roster operation.

```
-- Create a table containing upcoming 30 day's dates( used for duty roster )

DELIMITER $$

CREATE PROCEDURE thirty_days_from_now()
BEGIN
DECLARE cnt int;
DECLARE var_date date;
DROP TABLE IF EXISTS DATE_TEMP;
CREATE TABLE DATE_TEMP(
thirty_dates date NOT NULL
);
SET cnt = 1;
SET var_date = CURDATE();
WHILE (cnt<=30) DO
    INSERT INTO DATE_TEMP VALUES (var_date);
    SET var_date = DATE_ADD(var_date, INTERVAL 1 day);
    SET cnt = cnt + 1;
END WHILE;
END $$

--Query OK, 0 rows affected (0.02 sec)

DELIMITER ;
```

Triggers Used:

after_repairs_insert:

This trigger updates the Equipment table simultaneously with insertion of records in repairs.

```
CREATE TRIGGER after_repairs_insert
AFTER INSERT
ON Repairs FOR EACH ROW
BEGIN
UPDATE Equipment
SET Status = 'INREPAIR'
WHERE Equipment.EquipmentId = NEW.EquipmentId;
END $$

--Query OK, 0 rows affected (0.03 sec)
```

after_repairs_delete:

This trigger updates Equipment Table simultaneously with deletion of records in repairs.

```
CREATE TRIGGER after_repairs_delete
AFTER DELETE
ON Repairs FOR EACH ROW
BEGIN
UPDATE Equipment
SET Status = 'INUSE'
WHERE Equipment.EquipmentId = OLD.EquipmentId;
END $$

--Query OK, 0 rows affected (0.02 sec)
```

Indexes Used:

idx_login:

This index is created on LoginDetails, to speed up searching at login.

```
CREATE INDEX idx_login ON LoginDetails(Id, Passwd);

-- Query OK, 0 rows affected (0.07 sec)
-- Records: 0  Duplicates: 0  Warnings: 0
```

idx_gardener:

This index is created on the gardener table to increase searching speed of Id.

```
CREATE INDEX idx_gardener ON Gardener(Id);

-- Query OK, 0 rows affected (0.05 sec)
-- Records: 0  Duplicates: 0  Warnings: 0
```

idx_regions:

This index is created on the regions table to increase the searching speed of RegionCode.

```
CREATE INDEX idx_regions ON Regions(RegionCode);

-- Query OK, 0 rows affected (0.05 sec)
-- Records: 0  Duplicates: 0  Warnings: 0
```

idx_equip:

This index is created on the equipment table to increase the search speed of EquipmentId.

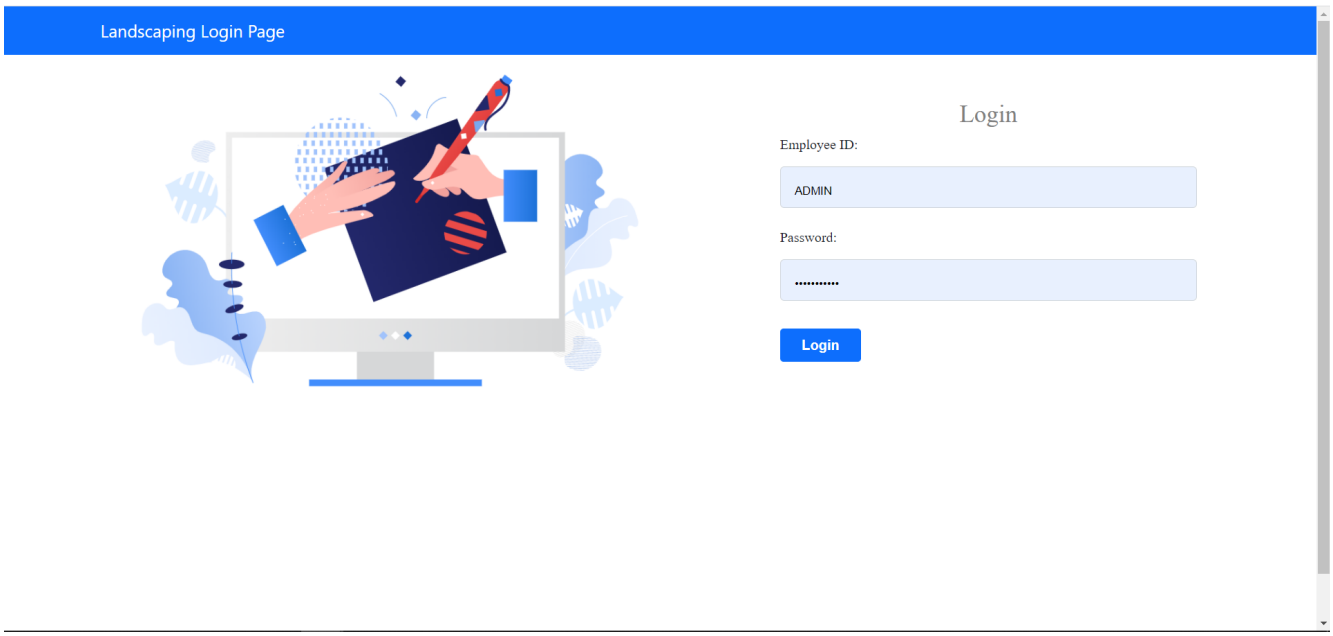
```
CREATE INDEX idx_equip ON Equipment(EquipmentId);

-- Query OK, 0 rows affected (0.04 sec)
-- Records: 0  Duplicates: 0  Warnings: 0
```

Screenshots:

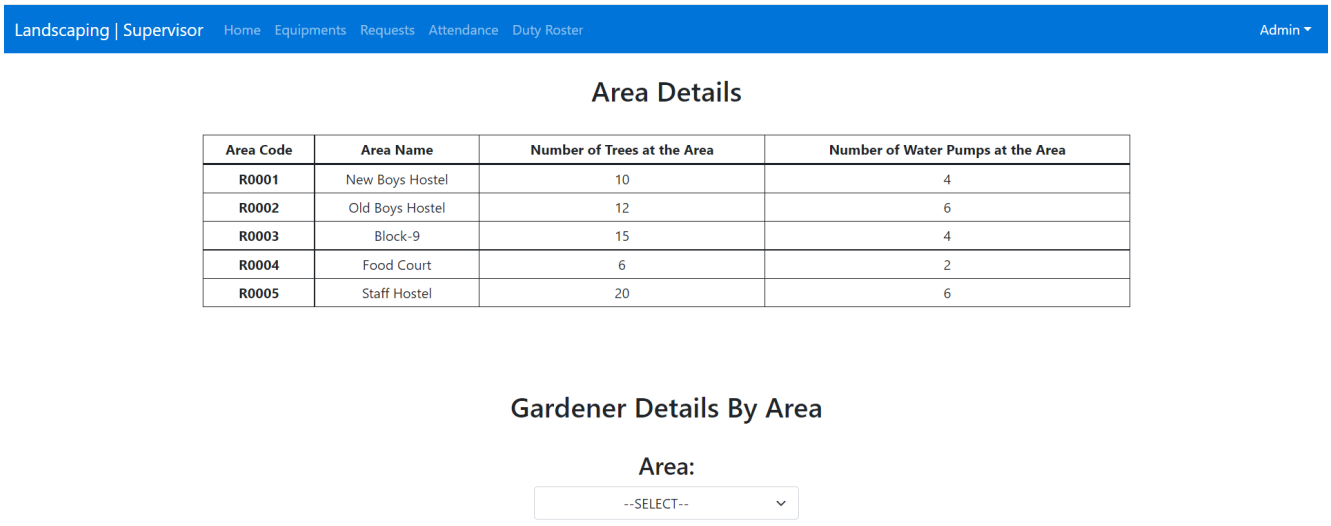
The screenshots/video are taken using small set of the dataset(mentioned in sql txt file) to ease printout and checking. Large Dataset is considered at the end.

Login Page:



The ID is ADMIN and Password is admin@admin.

Home Page:



This page gives the Area details and Gardener details by Area.

Area Code	Area Name	Number of Trees at the Area	Number of Water Pumps at the Area
R0001	New Boys Hostel	10	4
R0002	Old Boys Hostel	12	6
R0003	Block-9	15	4
R0004	Food Court	6	2
R0005	Staff Hostel	20	6

Gardener Details By Area

Area:

--SELECT--

R0001

ID	Name	Phone Number	Address	Add. Phone Number	Aadhar Number	Employment Type	Date of Joining	Start Date	End Date
A0001	Suresh Kumar	8888999910	A-102, Kampa Village, Bihta, Bihar	-	987654321712	P	2009-08-21	2021-06-21	2021-12-25
A0002	Mukesh Kumar	8888999912	A-100, Kampa Village, Bihta, Bihar	9578540000	987653421564	P	2010-04-22	2021-06-23	2021-12-25

On selecting Area(R0001), we get gardener details of that area(R0001).

Equipment Page:

Landscaping | Supervisor Home Equipments Requests Attendance Duty Roster Admin

Equipments under Repair

Equipment ID	Date Issued	Approx Dispatch Date	Remarks	Repair Done?
11	2021-11-15	2021-12-23	Blade broken	Repaired
18	2021-11-25	-	Hole in the fuel tank	Repaired

Equipments in Stock

Equipment ID	Equipment Name	Fuel Tank Capacity	Buying Date	Warranty Till Date	Previous Maintenance Date	Upcoming Maintenance Date	Status
11	Grasscutter X01	-	2010-08-21	2013-08-21	2021-07-21	2022-01-15	IN REPAIR
12	Grasscutter X02	-	2012-04-10	2014-04-10	2021-06-26	2022-01-02	IN USE
13	Grasscutter X03	-	2014-06-11	2016-06-11	2021-08-10	2022-02-10	IN USE
14	Hand Blower A01	0.58	2011-01-12	2016-01-12	2021-11-01	2022-05-02	IN USE
15	Hand Blower A02	0.75	2015-09-25	2016-09-25	2021-10-11	2022-04-09	IN USE
16	Hand Blower B02	0.49	2017-12-11	2020-12-11	2021-09-05	2022-03-01	IN USE

This page gives details about Equipments under Repair and Equipments in Stock.

17	Chain Saw Machine G1	0.56	2018-06-05	2021-06-05	2021-05-21	2022-12-10	IN USE
18	Chain Saw Machine G2	0.52	2020-06-05	2024-06-05	2021-09-27	2022-03-25	IN REPAIR
19	Chain Saw Machine G3	0.68	2021-10-23	2025-10-23	-	2022-04-20	IN USE
20	Shovel XSD	-	2008-02-14	-	-	-	IN USE

Submit Details for Equipments under Repair

Equipment ID:*

12

Approx. Dispatch Date:(Not Mandatory)

02-12-2021

Remarks:(Not Mandatory)

Axel Broken

Submit

This page also contains form for repair insertion.

Landscaping | Supervisor Home Equipments Requests Attendance Duty Roster Admin

Equipments under Repair

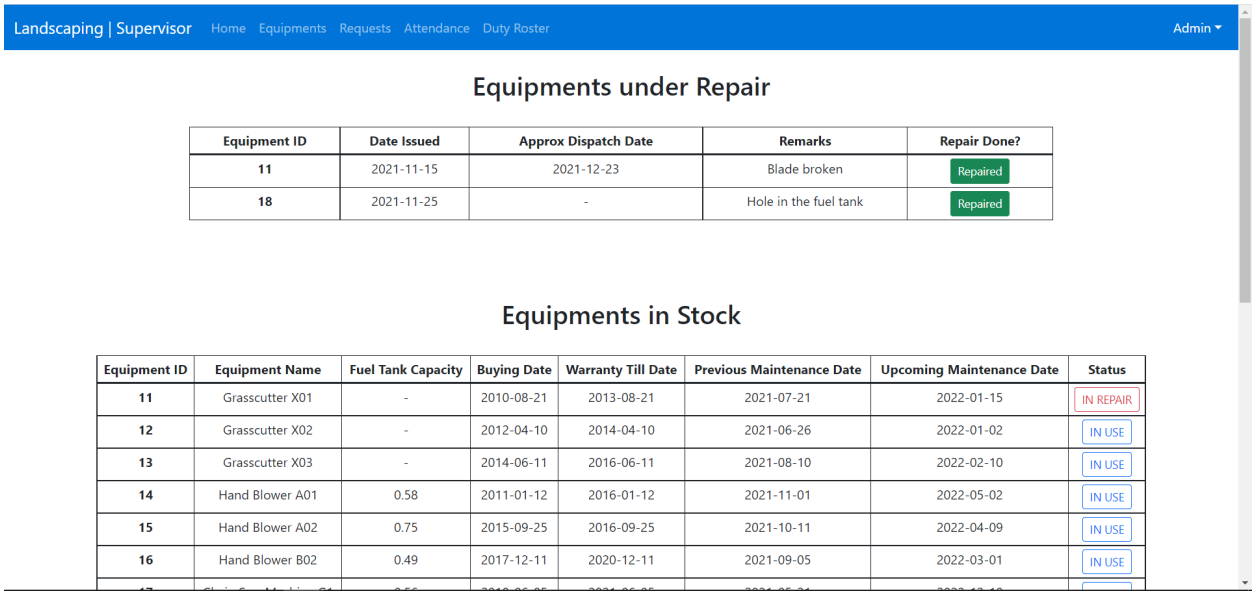
Equipment ID	Date Issued	Approx Dispatch Date	Remarks	Repair Done?
11	2021-11-15	2021-12-23	Blade broken	Repaired
18	2021-11-25	-	Hole in the fuel tank	Repaired
12	2021-11-30	2021-12-02	Axel Broken	Repaired

Equipments in Stock

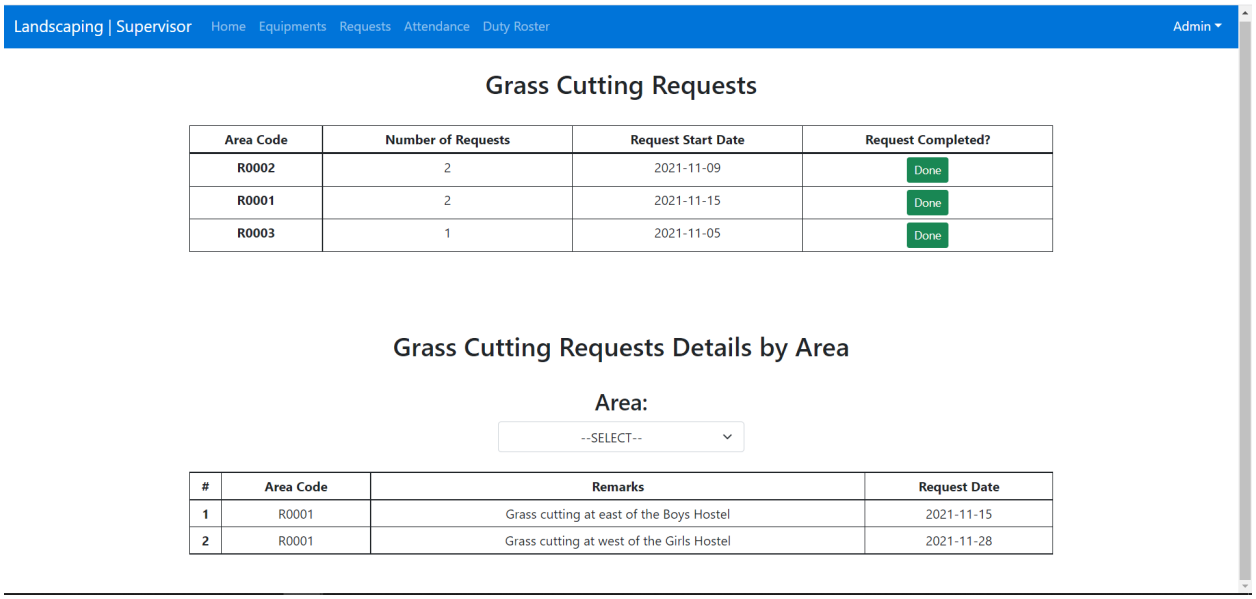
Equipment ID	Equipment Name	Fuel Tank Capacity	Buying Date	Warranty Till Date	Previous Maintenance Date	Upcoming Maintenance Date	Status
11	Grasscutter X01	-	2010-08-21	2013-08-21	2021-07-21	2022-01-15	IN REPAIR
12	Grasscutter X02	-	2012-04-10	2014-04-10	2021-06-26	2022-01-02	IN REPAIR
13	Grasscutter X03	-	2014-06-11	2016-06-11	2021-08-10	2022-02-10	IN USE
14	Hand Blower A01	0.58	2011-01-12	2016-01-12	2021-11-01	2022-05-02	IN USE
15	Hand Blower A02	0.75	2015-09-25	2016-09-25	2021-10-11	2022-04-09	IN USE

Here we can see on submitting the above form results in, addition of #ID = 12 element into repair table and status of the equipment(in Equipment Stock) is showing “IN REPAIR”.

On Clicking repaired button, it goes back into “INUSE” Status and is removed from the repair table.

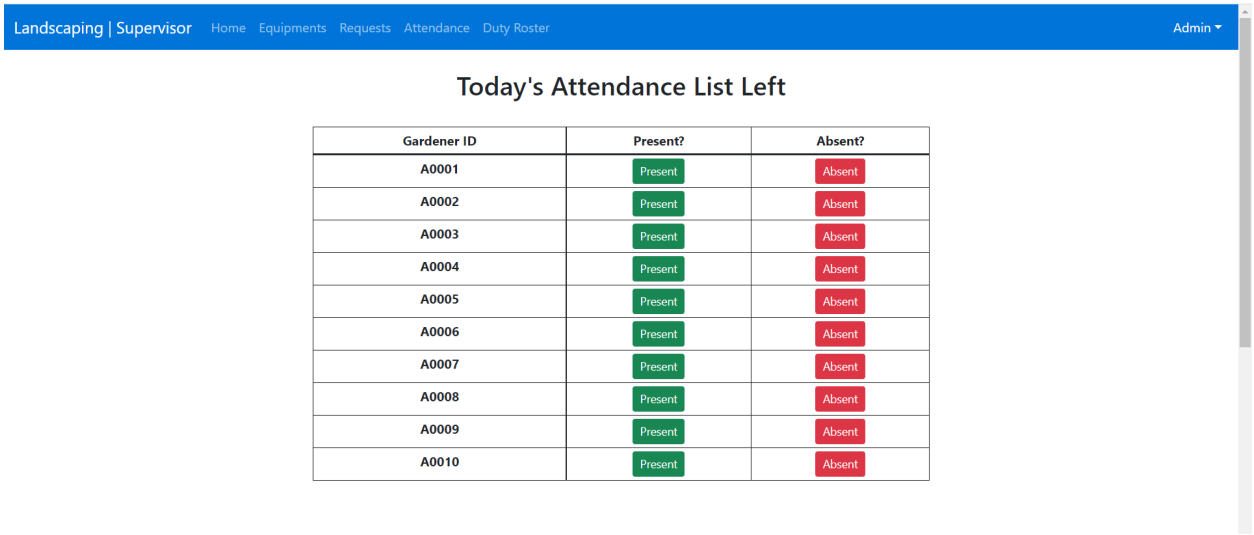


Requests Page:

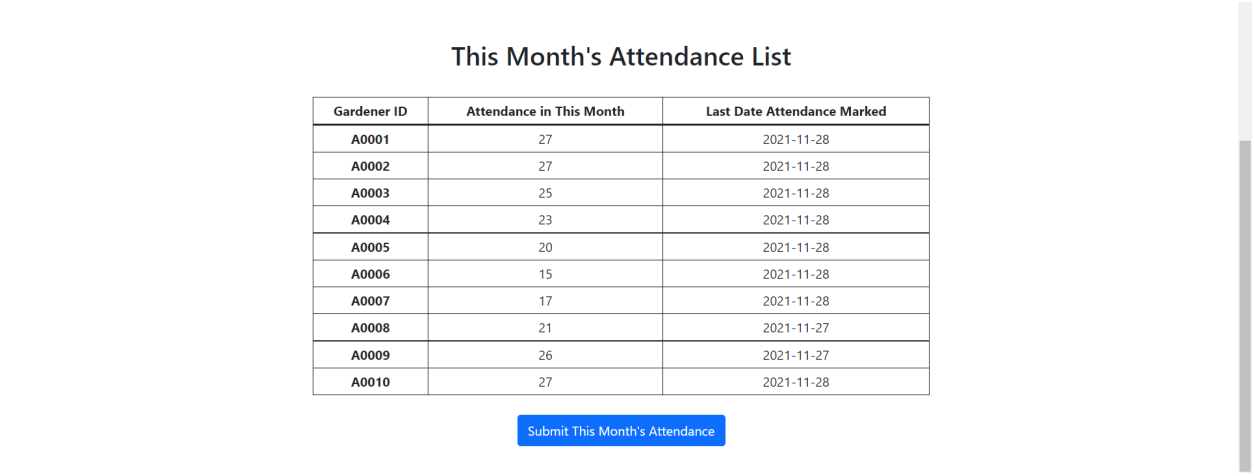


Here we can see all the grass cutting requests. The requests are grouped together and sorted firstly with Number of Requests in decreasing order and then secondly with oldest to newest Request Start Date. Clicking done will remove all the entries with that area code(will mark the completion of the request).
We can also see individual requests by area.

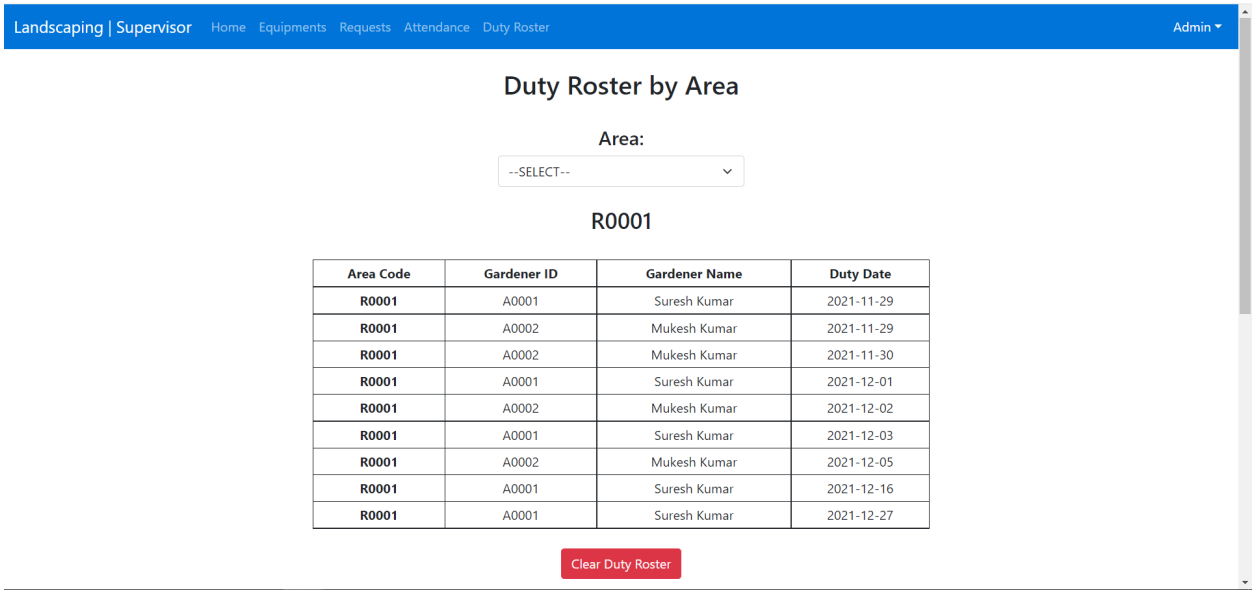
Attendance Page:



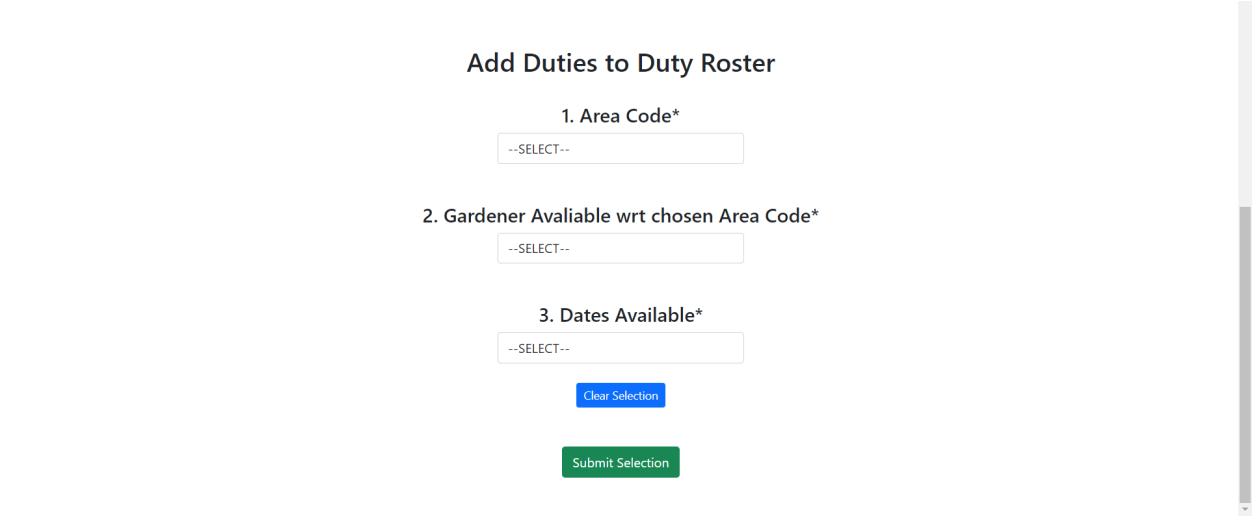
This page marks the attendance of the gardeners. It also shows total attendance in current month as well as last day, the attendance was marked. Also we can submit the whole list on the end of the month by clicking on Submit This Month's Attendance which stores the data into the log and resets the total attendance, as to start counting for the next month.



Duty Roster Page:

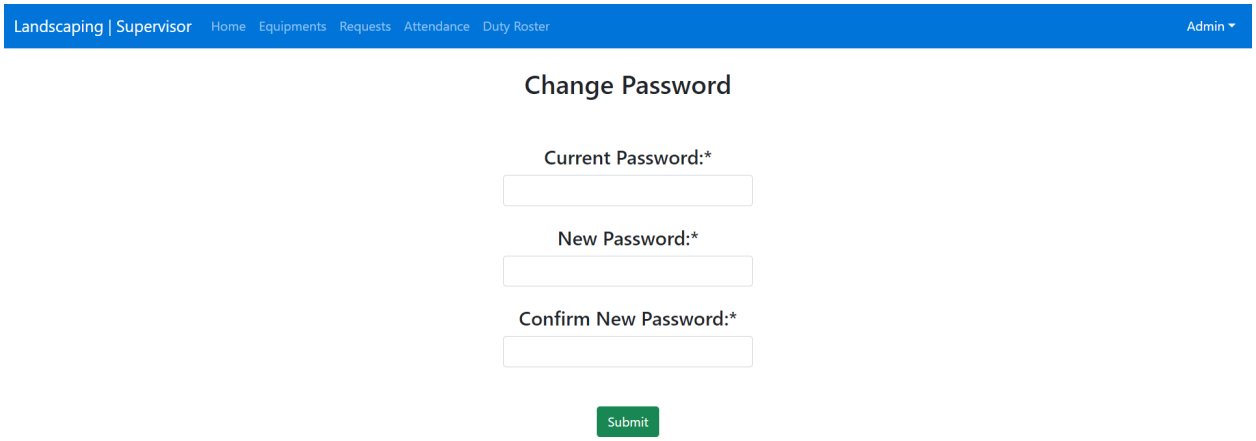


This page shows Duty Roster by Area. It also allows us to create a new duty roster.



Here the Gardener's availability is shown by Gardener Assignment Table which tells us which gardener is assign to which area. The Dates are shown by the selected Gardener's Available Dates(found by using thirty_days_from_now (~ 1 month) Procedure).

Change Password Page:



Here the new password is matched with confirm new password, if not same, the user is alerted, if same, the current password is checked, and if everything is correct then the password gets changed and the user is alerted for the same.

Size of the dataset:

Equipment Table:

```
SELECT COUNT(*) FROM Equipment;
+----------+
| COUNT(*) |
+----------+
|       40 |
+----------+
1 row in set (0.01 sec)
```

Gardener Table:

```
SELECT COUNT(*) FROM Gardener;
+----------+
| COUNT(*) |
+----------+
|       30 |
+----------+
1 row in set (0.00 sec)
```

Regions Table:

```
SELECT COUNT(*) FROM Regions;
+----------+
| COUNT(*) |
+----------+
|        5 |
+----------+
1 row in set (0.00 sec)
```

Repairs Table:

```
SELECT COUNT(*) FROM Repairs;
+----------+
| COUNT(*) |
+----------+
|        9 |
+----------+
1 row in set (0.00 sec)
```

Remaining tables are taken as given in the sql txt file with insert statements.

Other interesting queries which we can possibly include in the future is:

- **Use of Automatic Duty Roster generator**
- **Leave system among Gardeners**