CS355 Assignment Report

Team 22

Indian Institute of Technology, Patna

November 29, 2021

Module	**Contributor**	**Roll No**
Guest House Services	Arihant K. R	1901CS09
Market Services	Appaji	1901CS17
Landscaping Services	Abhishek Kumar	1901CS03

1 Market Services

1.1 Introduction

1.1.1 Background

In IIT Patna, There are several shops and canteens and they are currently managed manually i.e., paperwork, excel sheets etc... There are few drawbacks in current process :-

- Customer's feedback is not addressed properly.
- Extending Licenses doesn't necessarily depend on the shop's or shopkeeper's perfomance.
- Rent, Electricity payments are tracked manually, so that, penalties may not be effectively imposed.
- Lastly, there is no way to know list of current shop owners.

The objective of this project is to create an application to centrally manage all sorts of tasks related to Market Services, address above disadvantages and replace manual labour with a **delightful user experience**.

1.1.2 Functional Requirements

As a customer, User does following things :-

- Authenticate(Login/Register) himself/herself with a iitp email.
- View current shop owner's details and their license expiry details.
- Rate a Shop & Shopkeeper pair, and (may) give personalized feedback.
- Change their password, given current password, Update profile details.

As a Shopkeeper, User does following things :-

- Authenticate(Login/Register) himself/herself with an email.
- View current shop owner's details and their license expiry details.
- Pay his/her bills.
- View his/her past payments.
- Change their password, given current password, Update profile details.

As an Admin, User does following things :-

- Admin's account MUST be created from the database itself. Admin account cannot be created from the website, and *this is intentional*.
- Authenticate(Login/Register) himself/herself with an email.
- View current shop owner's details and their license expiry details.
- Extend a License expiry date.
- Approve/Decline Shopkeeper's payments.
- View Customer's rating & feedback.
- Change their password, given current password, Update profile details.

1.2 Database Design Decisions

- What is the purpose of **userId**? Why can't we use **email** as PRIMARY KEY?
  - Very valid point. Of course we can use. **email** is a VARCHAR(255), which is 255 bytes (at most). **userId** is INT, 4 bytes. Since they're to be used as PRIMARY KEY, these attributes are also used in other tables. We can easily optimize the space (255 - 4) = 251 bytes, by using **userId**. We can use UNIQUE constraint on **email** to avoid multiple users with same email.
- Why don't we have **feedbackId** similar to **userId/paymentId**?
  - It's not necessary to have a PRIMARY KEY on a table. If we have some queries which reference a particular row (or) other tables have foreign key referencing this table, presence of PRIMARY KEY is an advantage. In Feedbacks table, we don't have any of them, where as in Payments table, making a payment, approving a payment etc... correspond to one particular payment.

1.2.1 Utility function(s)

I'm using this function to confirm user role. It accepts two arguments, namely **userId** and **userRole**. This returns TRUE if the actual user's role matches with the argument.

```
CREATE FUNCTION `checkUserRole` (userId INT, userRole VARCHAR(10))
RETURNS BOOLEAN DETERMINISTIC
BEGIN
DECLARE result BOOLEAN DEFAULT 0;
SELECT 1 INTO result FROM `Users` WHERE `userId` = userId AND `userRole` = userRole LIMIT 1;
RETURN result;
END $$
```

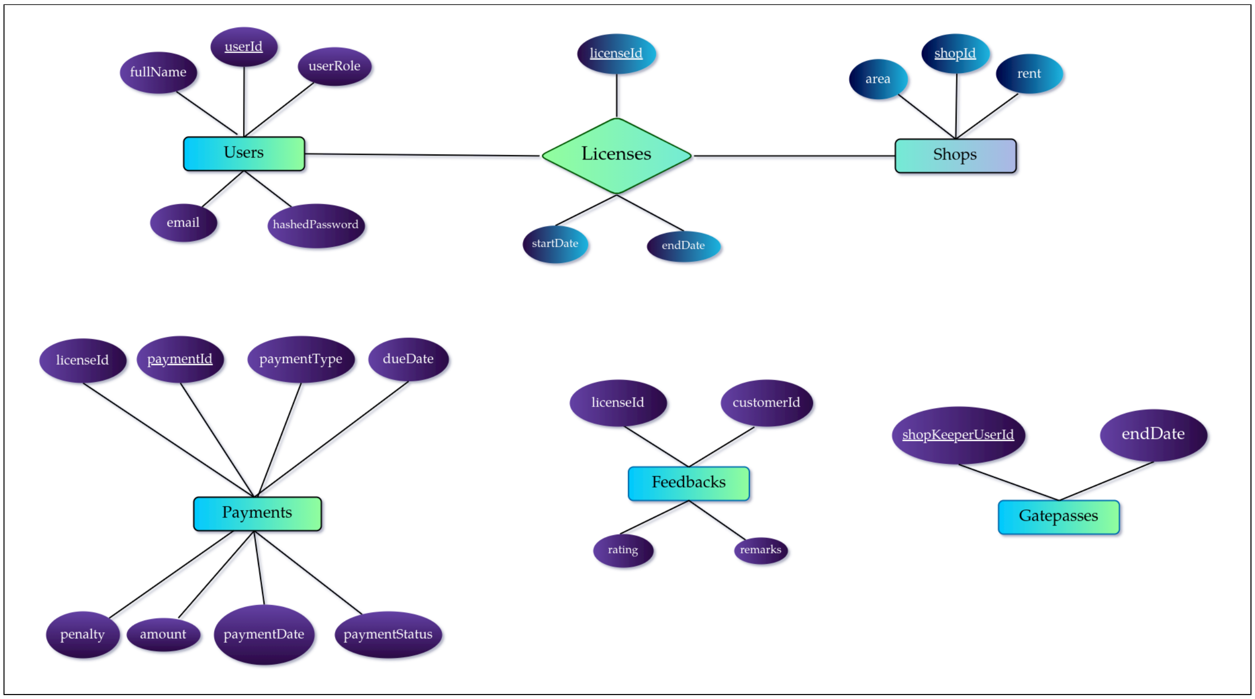
1.2.2 Constraints

Some not-so-obvious constraints are discussed here.

- **Users** table is assured to have a unique **email** with UNIQUE constraint. The PRIMARY KEY is auto-incremented however, and so, not controlled by the user. We'll check if the email is a valid email with a simple regular expression.
- **Gatepasses** have the gatepass details, ONLY for users with SHOPKEEPER role. Above **checkUserRole** function is used to confirm the users in this table are indeed SHOPKEEPERs.
- **Licenses** table binds a SHOPKEEPER and a SHOP. A BEFORE INSERT Trigger is used to confirm the user is SHOPKEEPER. **licenseId** is used as PRIMARY KEY, and it's auto-incremented.
- Requirement for **Payments** table is quite tricky. There has to be atmost 2 approved payments in a month (RENT, ELECTRICITY). UNIQUE can't be used directly on dueDate, Since, the constraint is on month, and type of payment as well. Here, We make use of functional keys.
  - UNIQUE KEY (`licenseId`, `paymentType`, (YEAR(`dueDate`)), (MONTH(`dueDate`)))

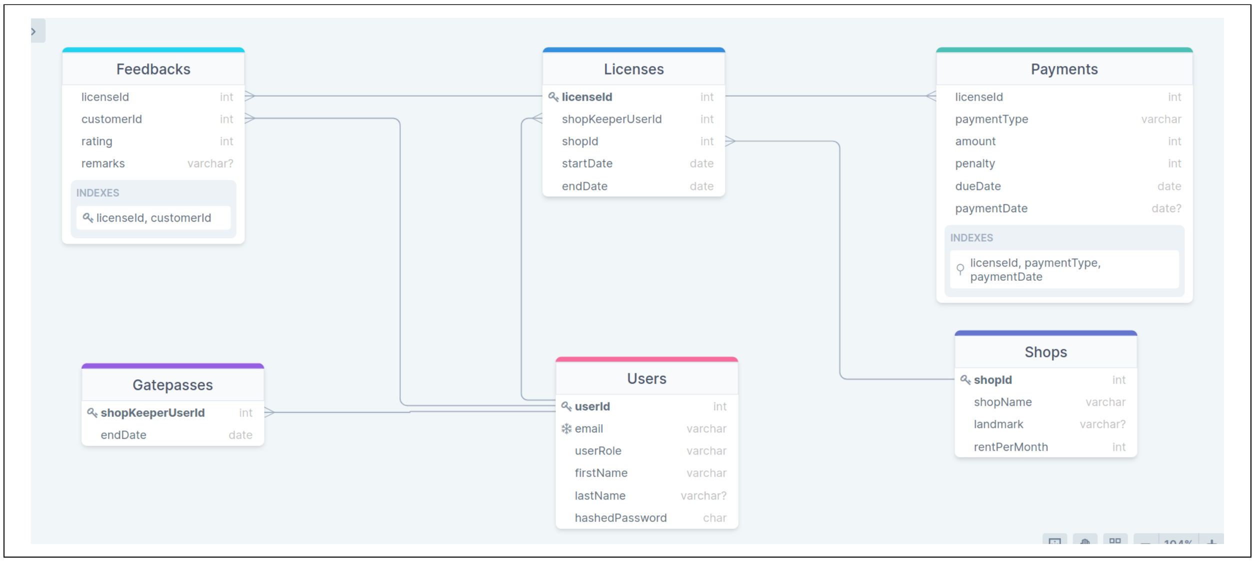
  PRIMARY KEY cannot be functional keys. So, this has to be UNIQUE KEY.

1.3 ER Diagram



ER Diagram depicting Entities and Relations

1.4 Relational Model



Feedbacks

licenseId	int
customerId	int
rating	int
remarks	varchar?

INDEXES

licenseId, customerId

Licenses

licenseId	int
shopKeeperUserId	int
shopId	int
startDate	date
endDate	date

Payments

licenseId	int
paymentType	varchar
amount	int
penalty	int
dueDate	date
paymentDate	date?

INDEXES

licenseId, paymentType, paymentDate

Gatepasses

shopKeeperUserId	int
endDate	date

Users

userId	int
email	varchar
userRole	varchar
firstName	varchar
lastName	varchar?
hashedPassword	char

Shops

shopId	int
shopName	varchar
landmark	varchar?
rentPerMonth	int

Relational Model depicting Entities and Relations

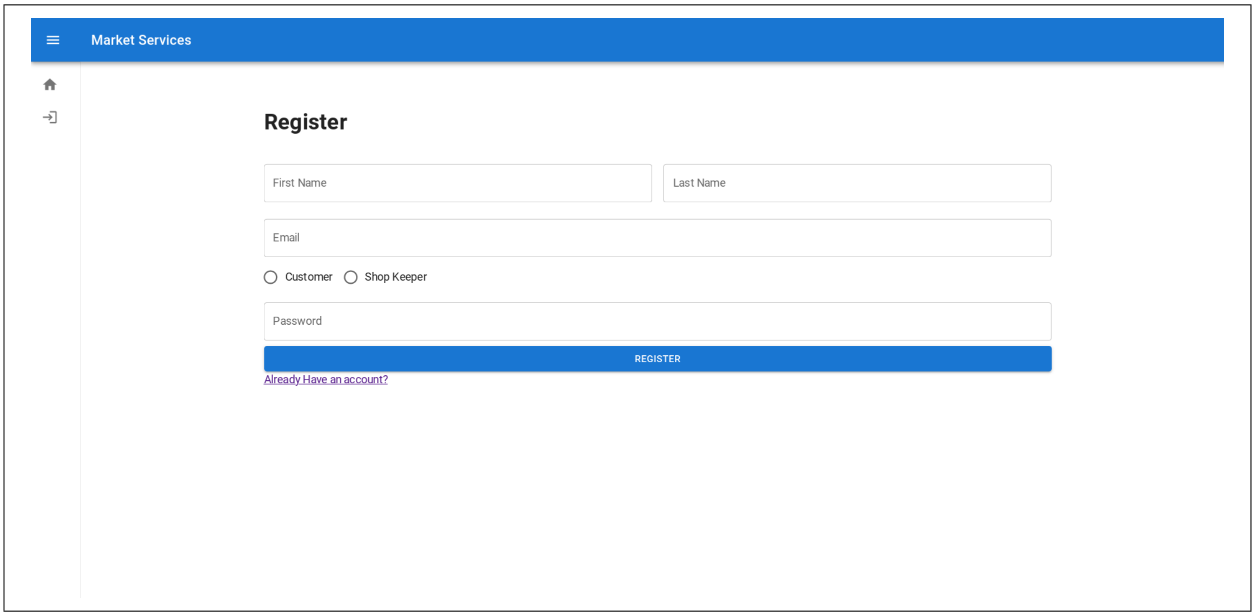
1.5 Creating Database

```sql
CREATE TABLE `Users`(
    `userId` INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    `email` VARCHAR(255) UNIQUE CHECK (`email` RLIKE '^\\S+@\\S+\\.\\S+$') NOT NULL,
    `userRole` VARCHAR(10) CHECK (`userRole` in ('CUSTOMER', 'SHOPKEEPER', 'ADMIN')) NOT
    ↪  NULL,
    `firstName` VARCHAR(30) NOT NULL,
    `lastName` VARCHAR(30),
    `hashedPassword` CHAR(60) NOT NULL
);

CREATE TABLE `Gatepasses`(
    `shopKeeperUserId` INT PRIMARY KEY NOT NULL,
    `endDate` DATE NOT NULL,
    FOREIGN KEY (`shopKeeperUserId`) REFERENCES `Users`(`userId`)
);

CREATE TABLE `Shops`(
    `shopId` INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    `shopName` VARCHAR(50) NOT NULL,
    `landmark` VARCHAR(100),
    `rentPerMonth` INT NOT NULL
);

CREATE TABLE `Licenses`(
    `licenseId` INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    `shopKeeperUserId` INT NOT NULL,
    `shopId` INT NOT NULL,
    `startDate` DATE NOT NULL,
    `endDate` DATE NOT NULL,
    FOREIGN KEY(`shopKeeperUserId`) REFERENCES `Users`(`userId`),
    FOREIGN KEY(`shopId`) REFERENCES `Shops`(`shopId`)
);

CREATE TABLE `Payments`(
    `paymentId` INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    `licenseId` INT NOT NULL,
    `paymentType` VARCHAR(15) CHECK (`paymentType` IN ('RENT', 'ELECTRICITY')) NOT NULL,
    `amount` INT NOT NULL,
    `penalty` INT NOT NULL DEFAULT 0,
    `dueDate` DATE NOT NULL,
    `paymentStatus` VARCHAR(20) CHECK(`paymentStatus` IN ('NOT PAID', 'PENDING APPROVAL',
    ↪  'PAID')) NOT NULL DEFAULT 'NOT PAID',
    `paymentDate` DATE,
    -- UNIQUE KEY because PRIMARY KEY doesn't support Functional Keys
    UNIQUE KEY (`licenseId`, `paymentType`, (YEAR(`dueDate`)), (MONTH(`dueDate`))),
    FOREIGN KEY(`licenseId`) REFERENCES `Licenses`(`licenseId`)
);
```

```
CREATE TABLE `Feedbacks`(
    `licenseId` INT NOT NULL,
    `customerId` INT NOT NULL,
    `rating` INT NOT NULL CHECK (`rating` in (-10, -5, 0, 5, 10)),
    `remarks` VARCHAR(200) NULL,
    PRIMARY KEY (`licenseId`, `customerId`),
    FOREIGN KEY(`licenseId`) REFERENCES `Licenses`(`licenseId`),
    FOREIGN KEY(`customerId`) REFERENCES `Users`(`userId`)
);

-- MySQL Function: Name is pretty much self-explanatory, i.e., checks user role.
DELIMITER $$
CREATE FUNCTION `checkUserRole` (userId INT, userRole VARCHAR(10))
RETURNS BOOLEAN DETERMINISTIC
BEGIN
DECLARE result BOOLEAN DEFAULT 0;
SELECT 1 INTO result FROM `Users` WHERE `userId` = userId AND `userRole` = userRole LIMIT 1;
RETURN result;
END $$

-- Gatepass should be issued only for ShopKeepers
CREATE TRIGGER `check_gatepass`
BEFORE INSERT ON `Gatepasses` FOR EACH ROW
BEGIN
IF NOT checkUserRole(NEW.shopKeeperUserId, 'SHOPKEEPER') THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Given user is NOT a ShopKeeper';
END IF;
END $$

-- License Constraints
CREATE TRIGGER `check_license`
BEFORE INSERT ON `Licenses` FOR EACH ROW
BEGIN
IF NOT checkUserRole(NEW.shopKeeperUserId, 'SHOPKEEPER') THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Given user is NOT a ShopKeeper';
END IF;
END $$

-- Feedback Constraints
CREATE TRIGGER `check_feedback`
BEFORE INSERT ON `Feedbacks` FOR EACH ROW
BEGIN
IF NOT checkUserRole(NEW.customerId, 'CUSTOMER') THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Given user is NOT a customer';
END IF;
END $$
DELIMITER ;
```

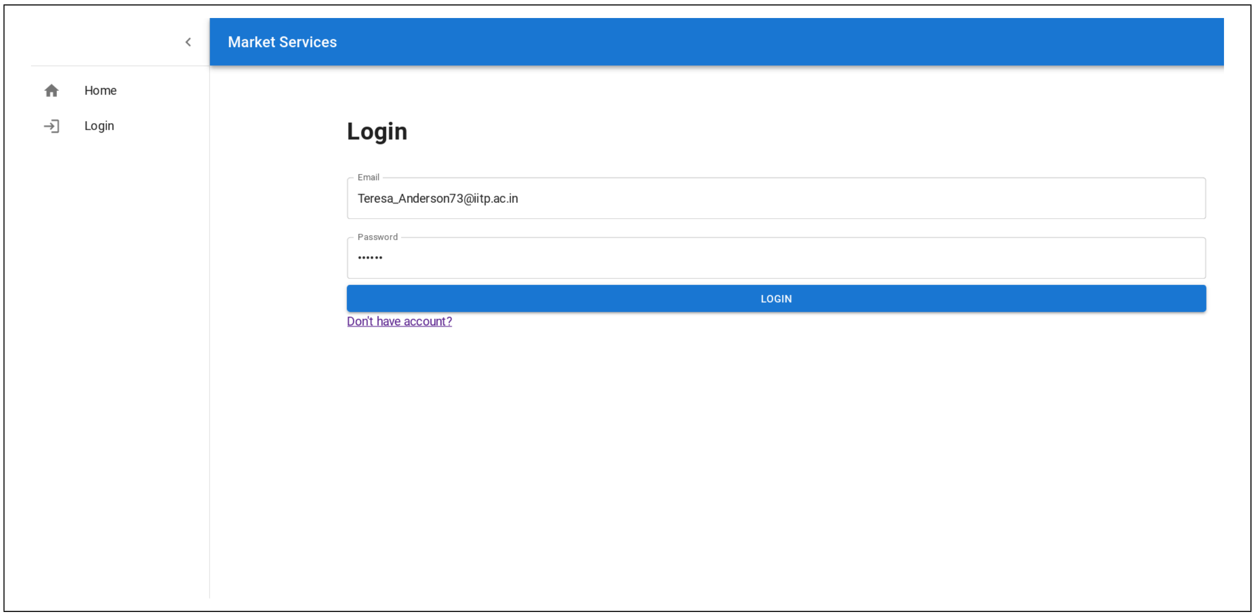
1.6 Web Pages

1.6.1 Register



New user registers here. Available roles are Customer and Shopkeeper. Admin role is NOT available here.

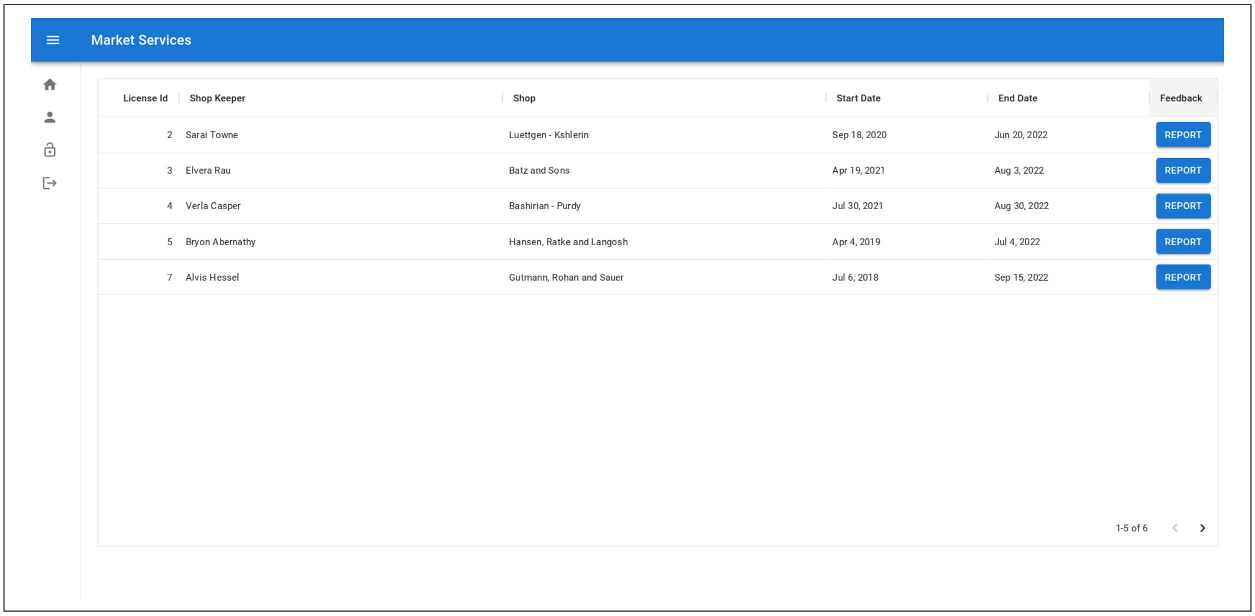
1.6.2 Login



Email: Teresa_Anderson73@iitp.ac.in

Password: ••••••

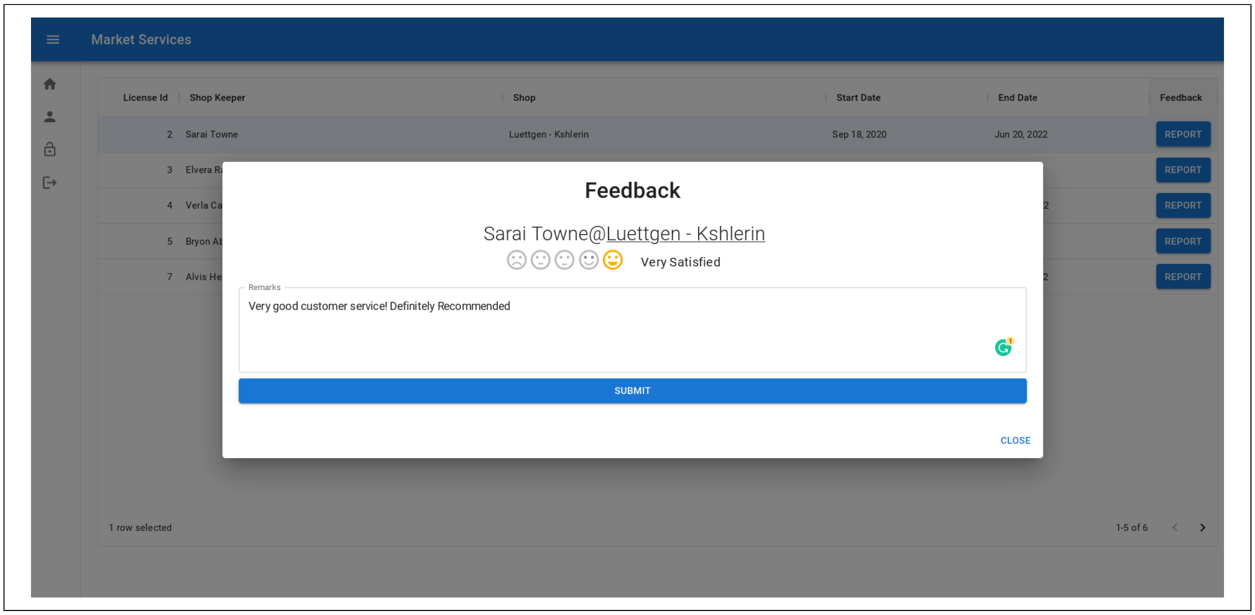
1.6.3 Licenses (for Customer)



License Id	Shop Keeper	Shop	Start Date	End Date	Feedback
2	Sarai Towne	Luettgen - Kshlerin	Sep 18, 2020	Jun 20, 2022	REPORT
3	Elvera Rau	Batz and Sons	Apr 19, 2021	Aug 3, 2022	REPORT
4	Verla Casper	Bashirian - Purdy	Jul 30, 2021	Aug 30, 2022	REPORT
5	Bryon Abernathy	Hansen, Ratke and Langosh	Apr 4, 2019	Jul 4, 2022	REPORT
7	Alvis Hessel	Gutmann, Rohan and Sauer	Jul 6, 2018	Sep 15, 2022	REPORT

The table is equipped with number of features such as sortable columns, Pagination, filter a column for a particular word, hide a column etc... We'll see these features in every upcoming page. In this page, **REPORT** button is appears only for **CUSTOMER**s.

1.6.4 Feedback by Customer



Feedback

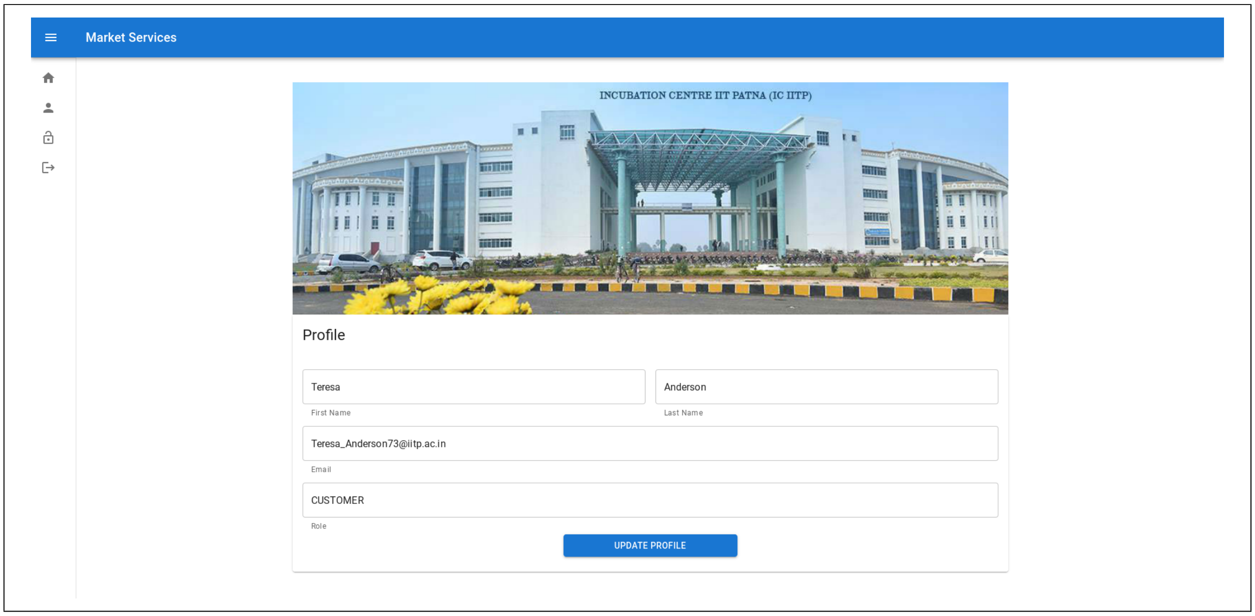
Sarai Towne@Luettgen - Kshlerin

Very Satisfied

Remarks: Very good customer service! Definitely Recommended

When **CUSTOMER** clicks report button, A dialog box is shown, and user can rate according to their experience. Remarks is preferred but not mandatory.

1.6.5 Profile



Profile

Teresa

First Name

Anderson

Last Name

Teresa_Anderson73@iitp.ac.in

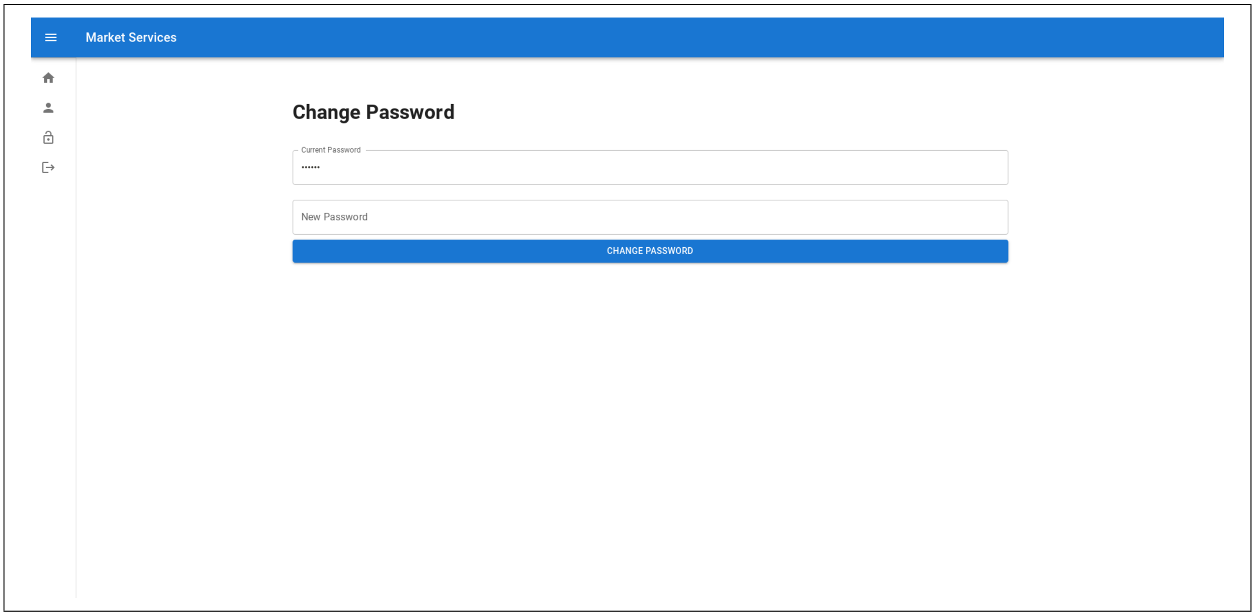
Email

CUSTOMER

Role

This page is same for all types of users. Users are presented with their details filled in a form. They can edit them and "update" their profile.

1.6.6 Change Password



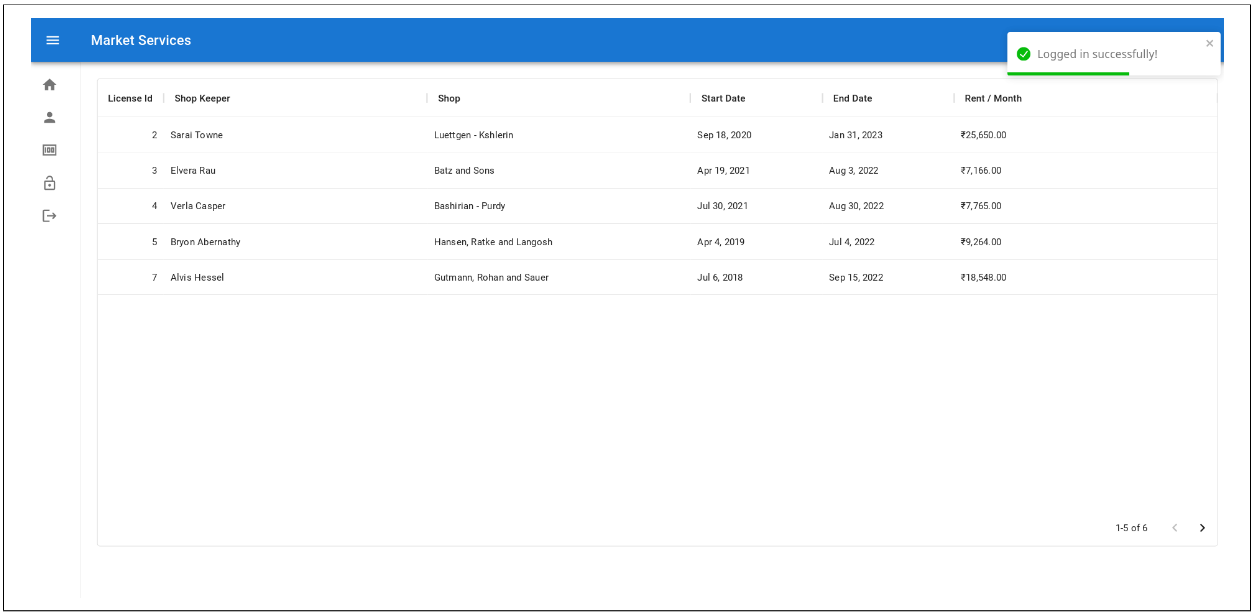
Change Password

Current Password

••••••

New Password

1.6.7 Licenses (for Shopkeeper)

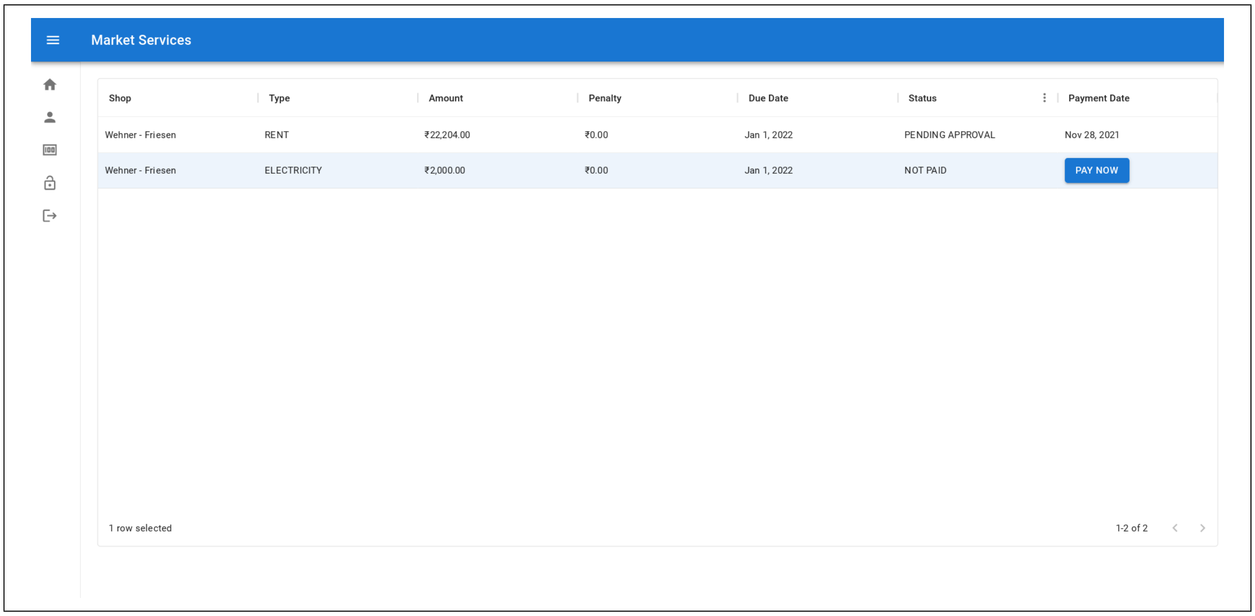


License Id	Shop Keeper	Shop	Start Date	End Date	Rent / Month
2	Sarai Towne	Luettgen - Kshlerin	Sep 18, 2020	Jan 31, 2023	₹25,650.00
3	Elvera Rau	Batz and Sons	Apr 19, 2021	Aug 3, 2022	₹7,166.00
4	Verla Casper	Bashirian - Purdy	Jul 30, 2021	Aug 30, 2022	₹7,765.00
5	Bryon Abernathy	Hansen, Ratke and Langosh	Apr 4, 2019	Jul 4, 2022	₹9,264.00
7	Alvis Hessel	Gutmann, Rohan and Sauer	Jul 6, 2018	Sep 15, 2022	₹18,548.00

1-5 of 6

As said, **REPORT** button is not shown to ShopeKeeper.

1.6.8 Payments (by Shopkeeper)



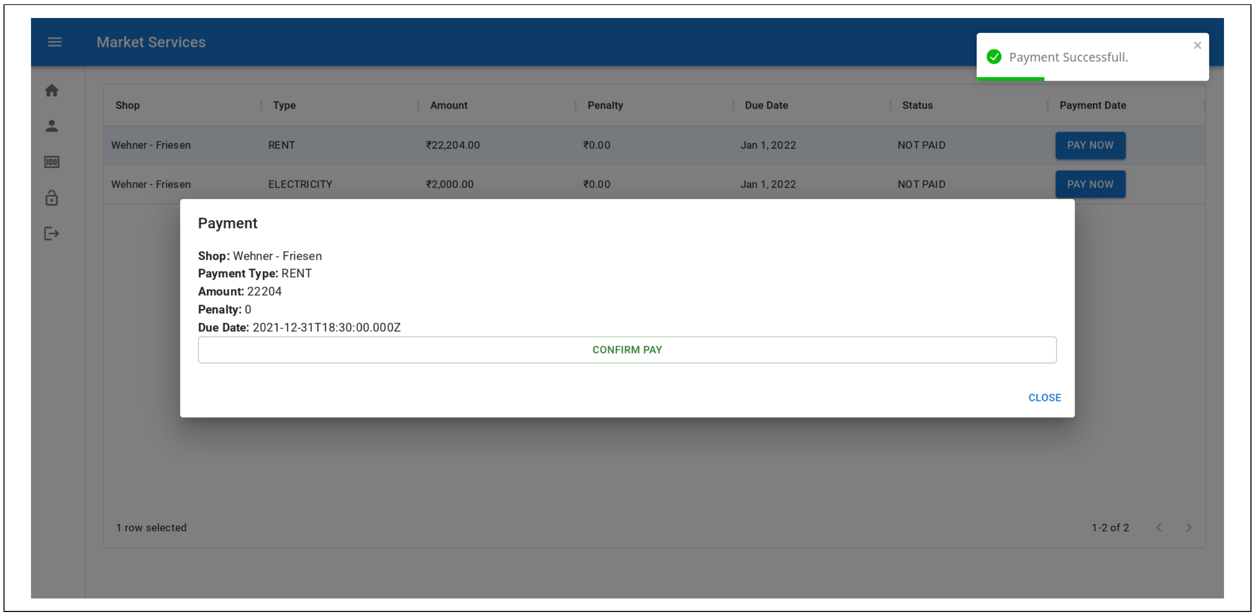
Shop	Type	Amount	Penalty	Due Date	Status	Payment Date
Wehner - Friesen	RENT	₹22,204.00	₹0.00	Jan 1, 2022	PENDING APPROVAL	Nov 28, 2021
Wehner - Friesen	ELECTRICITY	₹2,000.00	₹0.00	Jan 1, 2022	NOT PAID	PAY NOW

1 row selected

1-2 of 2

All payments corresponding to the logged in shopkeeper are shown in this page. Shopkeeper can view only his payments. If a payment is not yet made, A button is shown at the end of the row, which opens the following dialog box. Note that, A payment made by Shopkeeper is not done yet. Admint have to approve them.

1.6.9 Payment being made



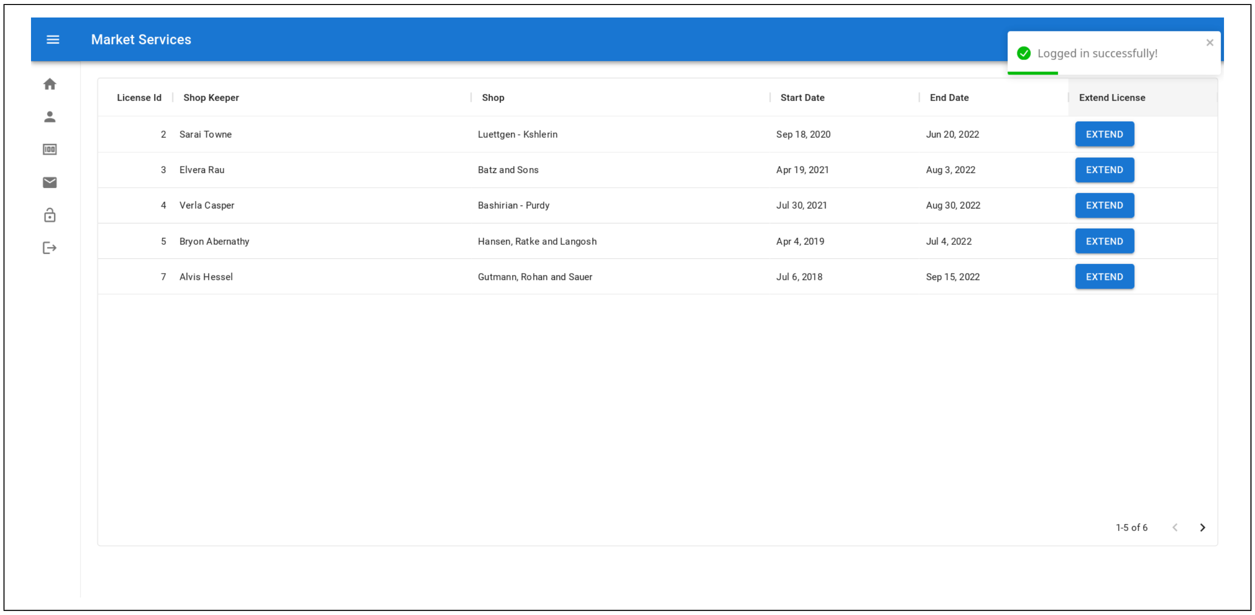
Shop	Type	Amount	Penalty	Due Date	Status	Payment Date
Wehner - Friesen	RENT	₹22,204.00	₹0.00	Jan 1, 2022	NOT PAID	PAY NOW
Wehner - Friesen	ELECTRICITY	₹2,000.00	₹0.00	Jan 1, 2022	NOT PAID	PAY NOW

Payment

Shop: Wehner - Friesen
Payment Type: RENT
Amount: 22204
Penalty: 0
Due Date: 2021-12-31T18:30:00.000Z

All details are shown to confirm by the ShopKeeper. Shopkeeper clicks **CONFIRM** to simulate the payment. An alert is shown at the top-right corner of the page.

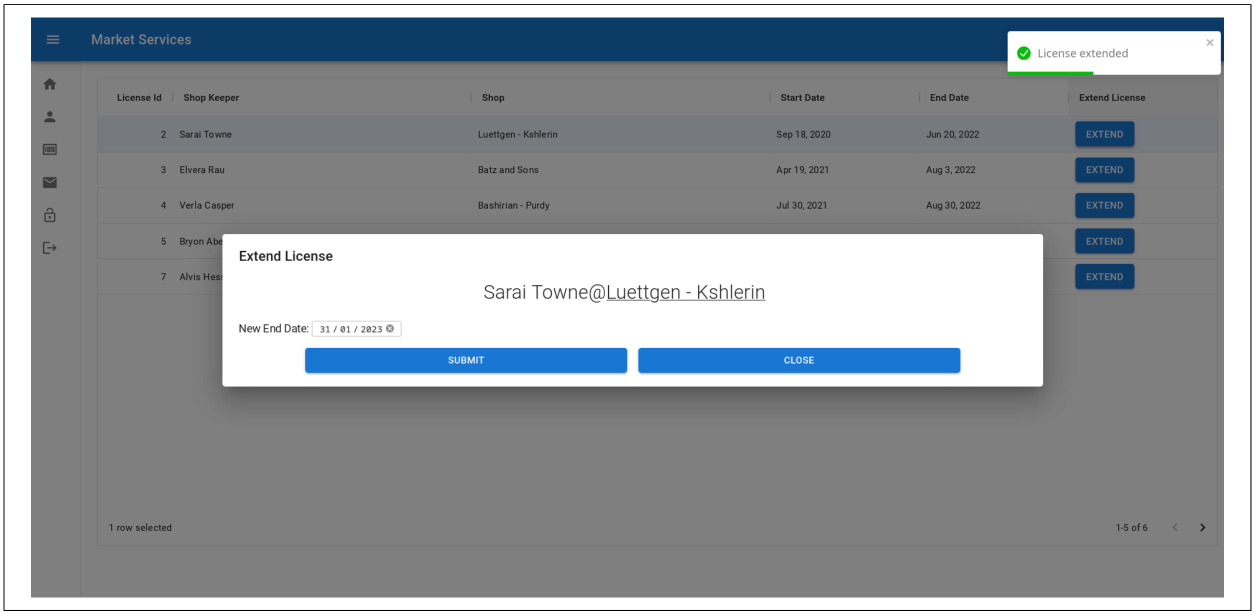
1.6.10 Licenses (Admin view)



License Id	Shop Keeper	Shop	Start Date	End Date	Extend License
2	Sarai Towne	Luettgen - Kshlerin	Sep 18, 2020	Jun 20, 2022	EXTEND
3	Elvera Rau	Batz and Sons	Apr 19, 2021	Aug 3, 2022	EXTEND
4	Verla Casper	Bashirian - Purdy	Jul 30, 2021	Aug 30, 2022	EXTEND
5	Bryon Abernathy	Hansen, Ratke and Langosh	Apr 4, 2019	Jul 4, 2022	EXTEND
7	Alvis Hessel	Gutmann, Rohan and Sauer	Jul 6, 2018	Sep 15, 2022	EXTEND

Admin cannot give feedback. Instead, Admin can extend the Licenses. We leave the judgement of which license can be extended to the admin. Below dialogbox opens up, after clicking extend.

1.6.11 Extend License



License extended

License Id	Shop Keeper	Shop	Start Date	End Date	Extend License
2	Sarai Towne	Luettgen - Kshlerin	Sep 18, 2020	Jun 20, 2022	EXTEND
3	Elvera Rau	Batz and Sons	Apr 19, 2021	Aug 3, 2022	EXTEND
4	Verla Casper	Bashirian - Purdy	Jul 30, 2021	Aug 30, 2022	EXTEND
5	Bryon Abe				EXTEND
7	Alvis Hes				EXTEND

1 row selected

1-5 of 6

Extend License

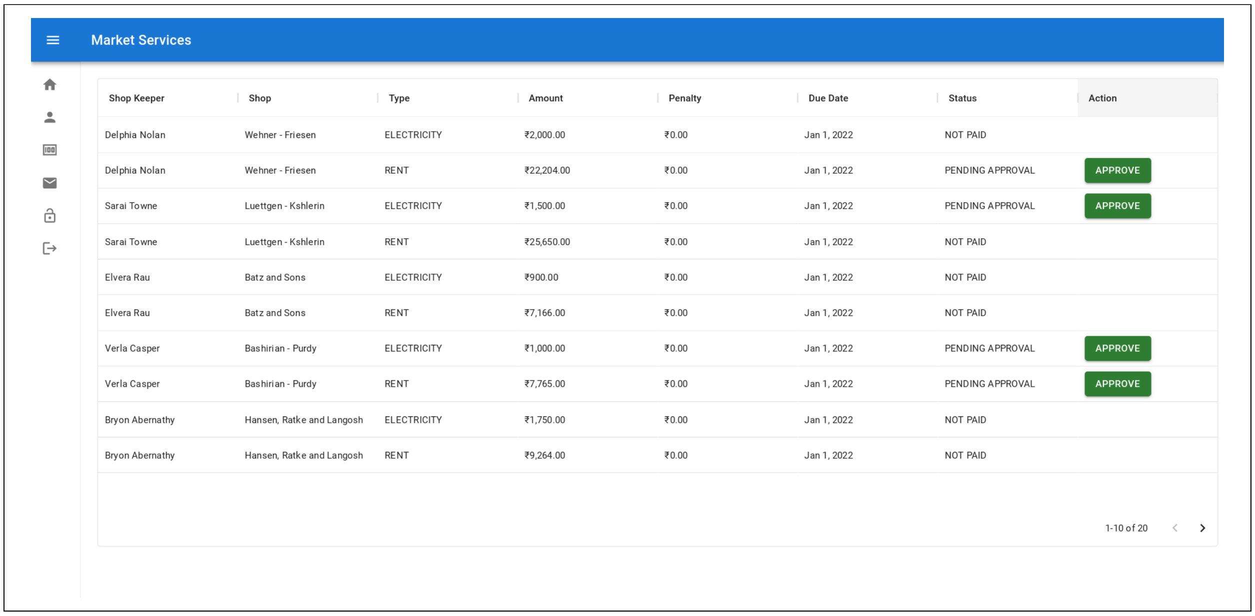
Sarai Towne@Luettgen - Kshlerin

New End Date: 31 / 01 / 2023

SUBMIT CLOSE

Shopkeeper and Shop names are shown and Admin is expected to give a new end date for license. An alert will be shown, after submitting the form.

1.6.12 Payments (Admin view)

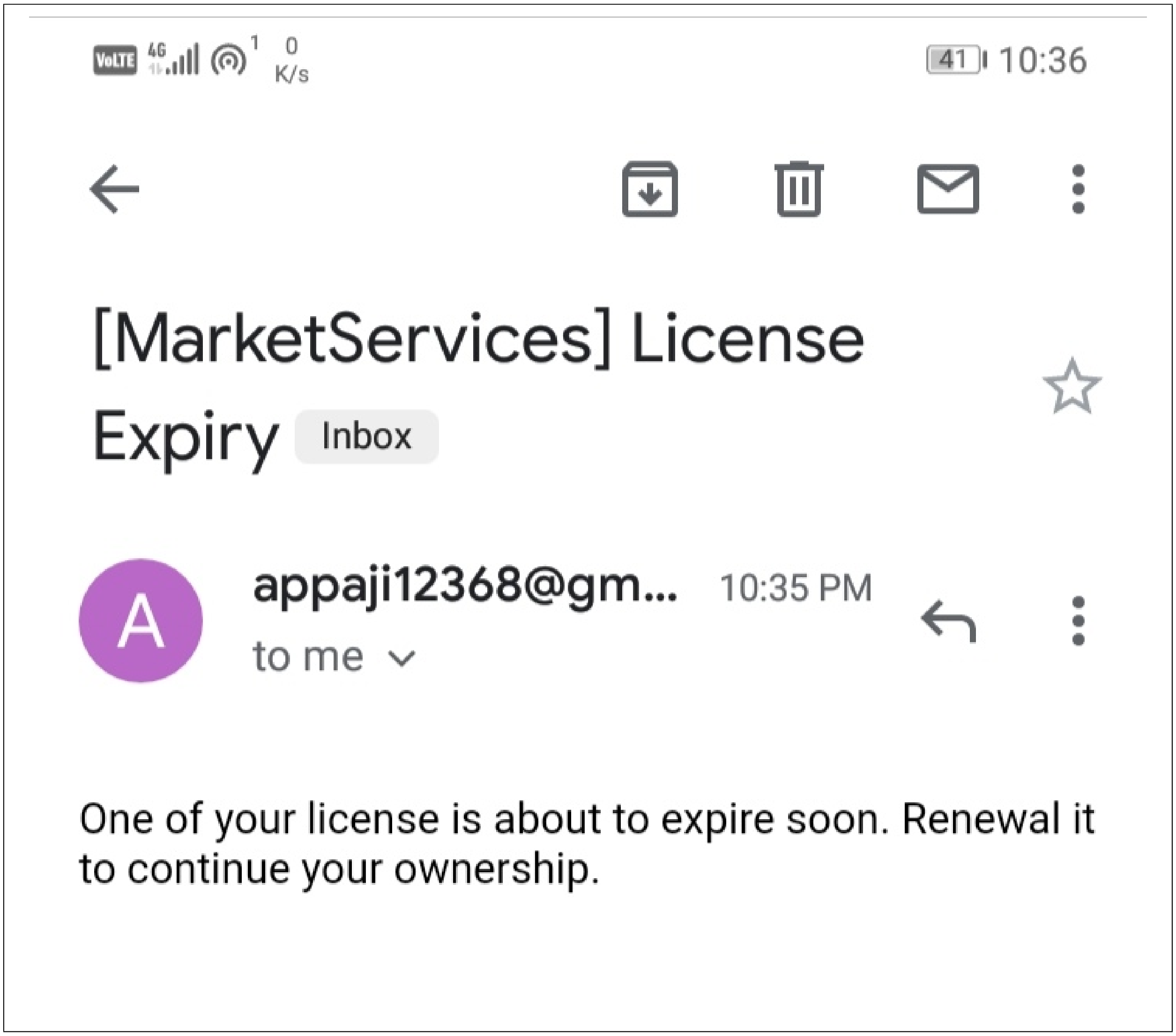


Shop Keeper	Shop	Type	Amount	Penalty	Due Date	Status	Action
Delphia Nolan	Wehner - Friesen	ELECTRICITY	₹2,000.00	₹0.00	Jan 1, 2022	NOT PAID	
Delphia Nolan	Wehner - Friesen	RENT	₹22,204.00	₹0.00	Jan 1, 2022	PENDING APPROVAL	APPROVE
Sarai Towne	Luettgen - Kshlerin	ELECTRICITY	₹1,500.00	₹0.00	Jan 1, 2022	PENDING APPROVAL	APPROVE
Sarai Towne	Luettgen - Kshlerin	RENT	₹25,650.00	₹0.00	Jan 1, 2022	NOT PAID	
Elvera Rau	Batz and Sons	ELECTRICITY	₹900.00	₹0.00	Jan 1, 2022	NOT PAID	
Elvera Rau	Batz and Sons	RENT	₹7,166.00	₹0.00	Jan 1, 2022	NOT PAID	
Verla Casper	Bashirian - Purdy	ELECTRICITY	₹1,000.00	₹0.00	Jan 1, 2022	PENDING APPROVAL	APPROVE
Verla Casper	Bashirian - Purdy	RENT	₹7,765.00	₹0.00	Jan 1, 2022	PENDING APPROVAL	APPROVE
Bryon Abernathy	Hansen, Ratke and Langosh	ELECTRICITY	₹1,750.00	₹0.00	Jan 1, 2022	NOT PAID	
Bryon Abernathy	Hansen, Ratke and Langosh	RENT	₹9,264.00	₹0.00	Jan 1, 2022	NOT PAID	

1-10 of 20

Admin can view all(past & present) payments in this page. For convenience, Admin can filter **PENDING APPROVAL** from the Status column, and approve each payment.

1.6.13 License Reminder



[MarketServices] License Expiry Inbox

appaji12368@gm... 10:35 PM

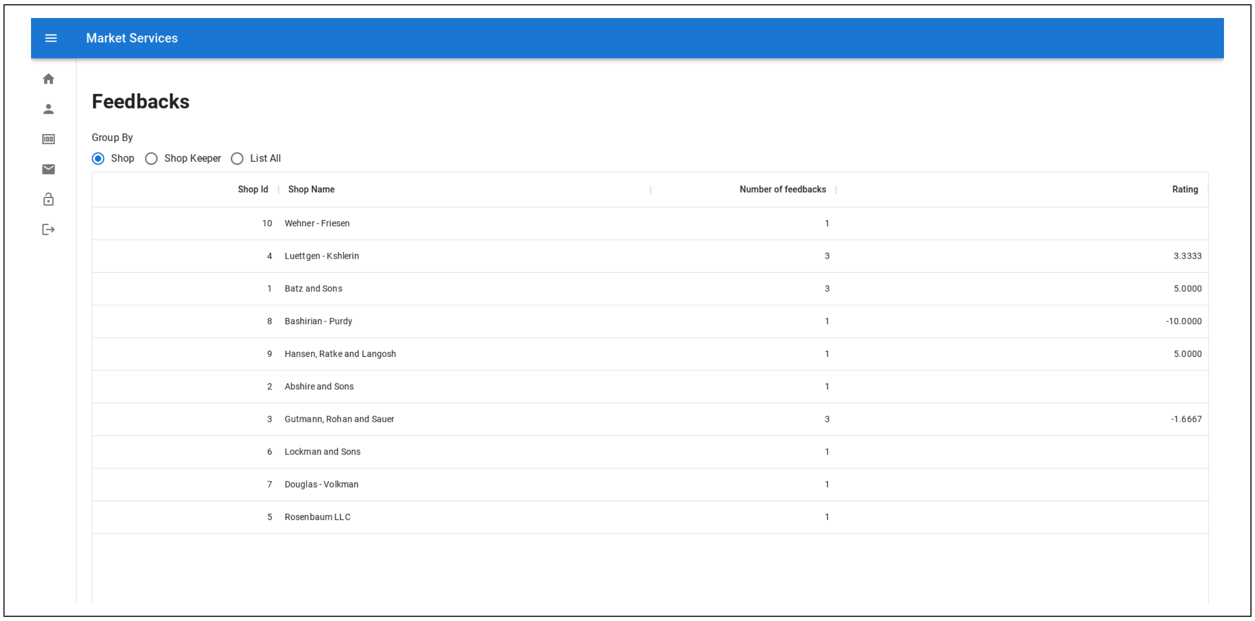
to me

One of your license is about to expire soon. Renewal it to continue your ownership.

There is a Node.js script(**Reminder.mjs**), which calls a function periodically to do following things.

- Get the list of current active licenses.
- Check if any license is expiring within 30 days.
- Send a mail to the owner of the license.

1.6.14 Feedbacks (Grouped by Shops)



Feedbacks

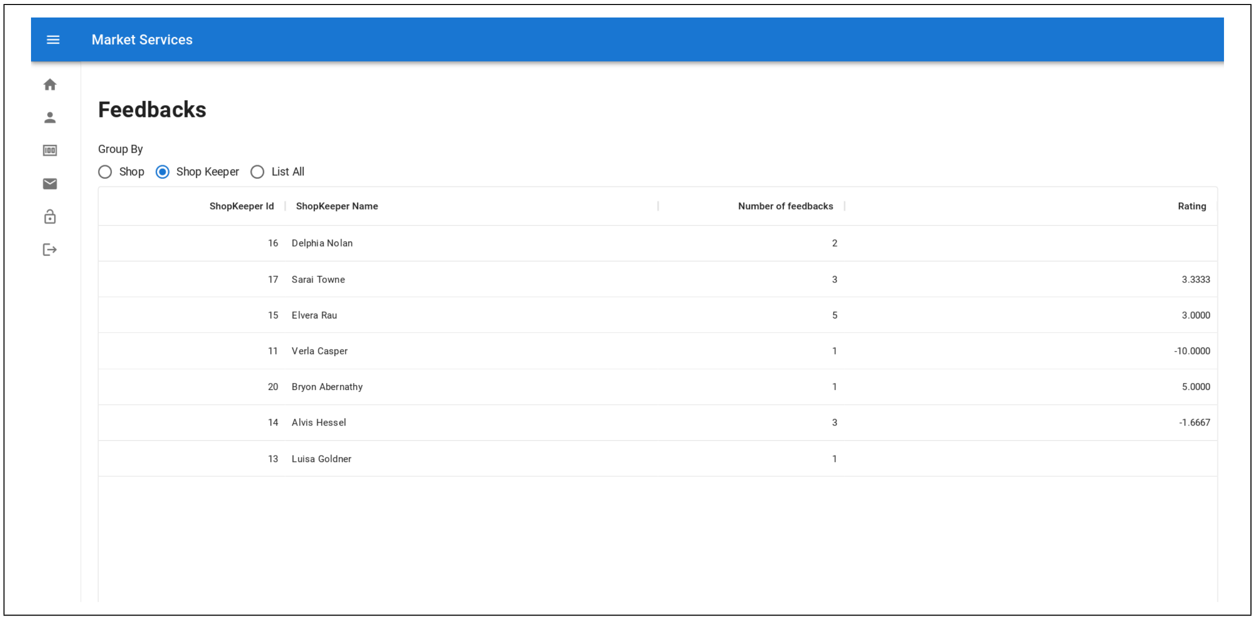
Group By

● Shop ○ Shop Keeper ○ List All

Shop Id	Shop Name	Number of feedbacks	Rating
10	Wehner - Friesen	1	
4	Luettgen - Kshlerin	3	3.3333
1	Batz and Sons	3	5.0000
8	Bashirian - Purdy	1	-10.0000
9	Hansen, Ratke and Langosh	1	5.0000
2	Abshire and Sons	1	
3	Gutmann, Rohan and Sauer	3	-1.6667
6	Lockman and Sons	1	
7	Douglas - Volkman	1	
5	Rosenbaum LLC	1	

Feedbacks are grouped by shops. Average rating, Number of feedbacks is shown for each **SHOP**.

1.6.15 Feedbacks (Grouped by Shopkeeper)



Feedbacks

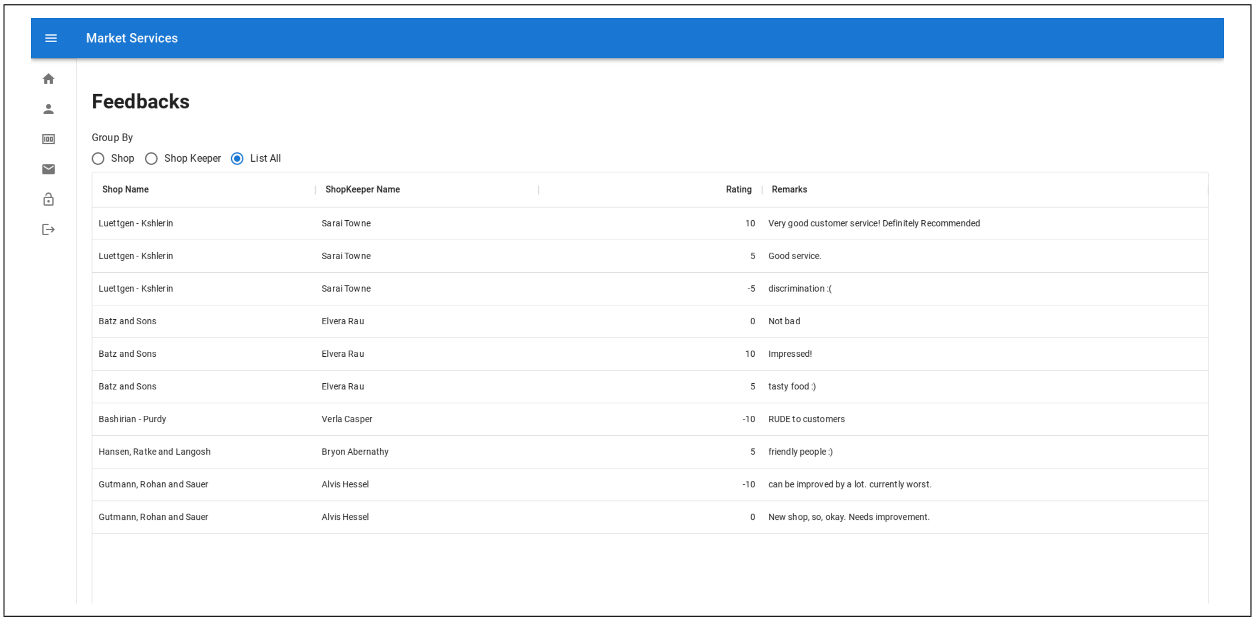
Group By

○ Shop ● Shop Keeper ○ List All

ShopKeeper Id	ShopKeeper Name	Number of feedbacks	Rating
16	Delphia Nolan	2	
17	Sarai Towne	3	3.3333
15	Elvera Rau	5	3.0000
11	Verla Casper	1	-10.0000
20	Bryon Abernathy	1	5.0000
14	Alvis Hessel	3	-1.6667
13	Luisa Goldner	1	

Feedbacks are grouped by Shopkeeper. Average rating, Number of feedbacks is shown for each **SHOPKEEPER**.

1.6.16 Feedbacks (List view)



Feedbacks

Group By

○ Shop ○ Shop Keeper ● List All

Shop Name	ShopKeeper Name	Rating	Remarks
Luettgen - Kshlerin	Sarai Towne	10	Very good customer service! Definitely Recommended
Luettgen - Kshlerin	Sarai Towne	5	Good service.
Luettgen - Kshlerin	Sarai Towne	-5	discrimination :(
Batz and Sons	Elvera Rau	0	Not bad
Batz and Sons	Elvera Rau	10	Impressed!
Batz and Sons	Elvera Rau	5	tasty food :)
Bashirian - Purdy	Verla Casper	-10	RUDE to customers
Hansen, Ratke and Langosh	Bryon Abernathy	5	friendly people :)
Gutmann, Rohan and Sauer	Alvis Hessel	-10	can be improved by a lot. currently worst.
Gutmann, Rohan and Sauer	Alvis Hessel	0	New shop, so, okay. Needs improvement.

All feedbacks are shown in list view here. Remarks are included in this view. Admin can filter certain words from the table

1.7 Sample SQL Queries

1.7.1 Current shop details of different areas of the campus

```sql
SELECT CONCAT_WS(' ', firstName, lastName) AS shopKeeper,
shopName, landmark, rentPerMonth as rent, startDate, endDate
FROM Licenses
INNER JOIN Users ON Licenses.shopKeeperUserId = Users.userId
INNER JOIN Shops ON Licenses.shopId = Shops.shopId
WHERE startDate <= CURDATE() AND CURDATE() <= endDate;
```

1.7.2 Details of shop keepers and their security pass validity

```sql
SELECT * FROM Users
JOIN Gatepasses ON Users.userRole = 'SHOPKEEPER'
AND Gatepasses.shopKeeperUserId = Users.userId;
```

1.7.3 Reminders for expiring license agreement period

```sql
SELECT * FROM Licenses
JOIN Users ON Licenses.shopKeeperUserId = Users.userId
AND startDate <= CURDATE() AND CURDATE() <= endDate
WHERE DATEDIFF(endDate, CURDATE()) <= 30;
```

1.7.4 Pending charges from each shop

```sql
SELECT Licenses.shopId, Shops.shopName,
SUM(amount) as 'Pending Charges' FROM Payments
JOIN Licenses ON Payments.paymentStatus != 'PAID'
AND Payments.licenseId = Licenses.licenseId
JOIN Shops ON Licenses.shopId = Shops.shopId
GROUP BY shopId;
```

1.7.5 Summary of performances of each shop

```sql
SELECT Licenses.shopId, shopName, COUNT(*) as count,
IFNULL(SUM(rating), NULL) / COUNT(*) as rating
FROM Licenses
LEFT JOIN Feedbacks ON Licenses.licenseId = Feedbacks.licenseId
LEFT JOIN Users ON Licenses.shopKeeperUserId = Users.userId
LEFT JOIN Shops ON Licenses.shopId = Shops.shopId GROUP BY shopId
```

1.8 Deployement

In this project, I clearly decoupled backend and frontend into independent services. So, they can be deployed independently. Below is the bash script to deploy backend on a Linux server.

1.8.1 Backend

```bash
#!/bin/bash

PKG_MANAGER="yarn"

if ! command -v node &> /dev/null
then
    echo "[-] Node.JS isn't installed."
    exit 1
else
    echo "[+] Node.JS found"
fi

if ! command -v $PKG_MANAGER &> /dev/null
then
    echo "[-] Yarn isn't installed. Switching to npm..."
    PKG_MANAGER="npm"
else
    echo "[+] Yarn is set as package manager."
fi

$PKG_MANAGER install;
sudo systemctl start mysql;
node server.js;
```

1.8.2 Frontend

Update Server URL in **.env**, to backend's URL (probably localhost:8000 or something else if it's on production). This way, we connect frontend to receive data from backend.

```
yarn build
```

Above command prepares a **build** directory, which can be hosted via a CDN or static-site hosting service.

1.9 Source Code

Here's the link to my GitHub Repository: https://github.com/CITIZENDOT/MarketServices